DISEÑO DE UN PROTOTIPO DE APLICACIÓN MÓVIL PARA CIFRADO DE MENSAJES SOBRE PLATAFORMA ANDROID

GIOVANI NEIRA CASTELLANOS

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA-ECBTI
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
VILLAVICENCIO
2019

DISEÑO DE UN PROTOTIPO DE APLICACIÓN MÓVIL PARA CIFRADO DE MENSAJES SOBRE PLATAFORMA ANDROID

GIOVANI NEIRA CASTELLANOS

Proyecto de grado para optar el título de especialista en seguridad informática

Asesor:
Gabriel Alberto Puerta Aponte
Ingeniero en informática
Especialista en seguridad informática
MBA

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA-ECBTI
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
VILLAVICENCIO
2019

Nota de aceptación:

_____
_____
_____
_____
_____
_____

_____
Presidente del Jurado

_____
Jurado

_____
Jurado

Villavicencio, noviembre de 2019

AGRADECIMIENTOS

Agradezco a todas las personas que con su apoyo moral y económico contribuyeron con este sueño, a mi esposa carolina incondicional siempre con una voz de aliento en los momentos difíciles, a mi gran amigo Juan Fajardo, ingeniero de sistemas mano derecha que con sus conocimientos me dieron mucha orientación para culminar mi especialización.

A la universidad nacional abierta y a distancia por acogerme en su alma mater.

A mi tutor y directores de trabajo, por su orientación y confianza en el desarrollo y culminación de este proyecto

TABLA DE CONTENIDO

Pág.

LISTA DE FIGURAS

Pág.

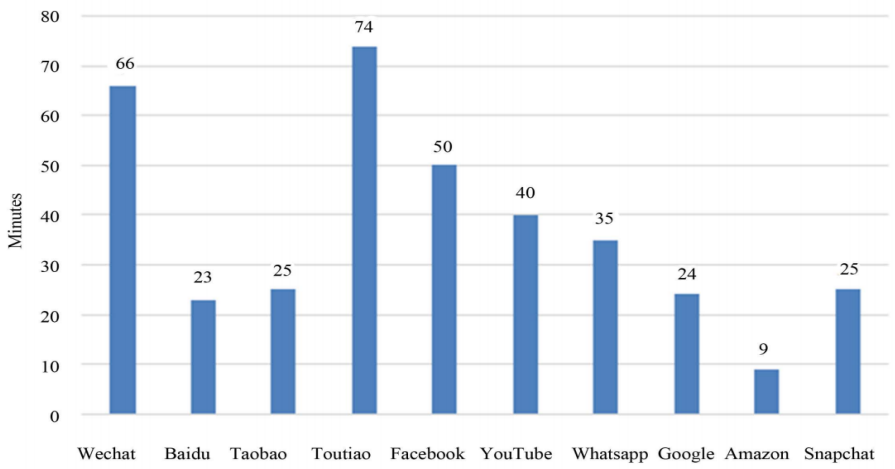
LISTA DE TABLAS

INTRODUCCIÓN

Debido al rápido crecimiento, “el desarrollo de internet y las comunicaciones móviles, ha hecho que el tiempo promedio que se emplea en las aplicaciones móviles en los Estados Unidos ha llegado a 2 horas y 15 minutos por día. El uso de aplicaciones en Corea del sur, Brasil y México ha alcanzado 3 horas”. (Tao, K. ,2018)

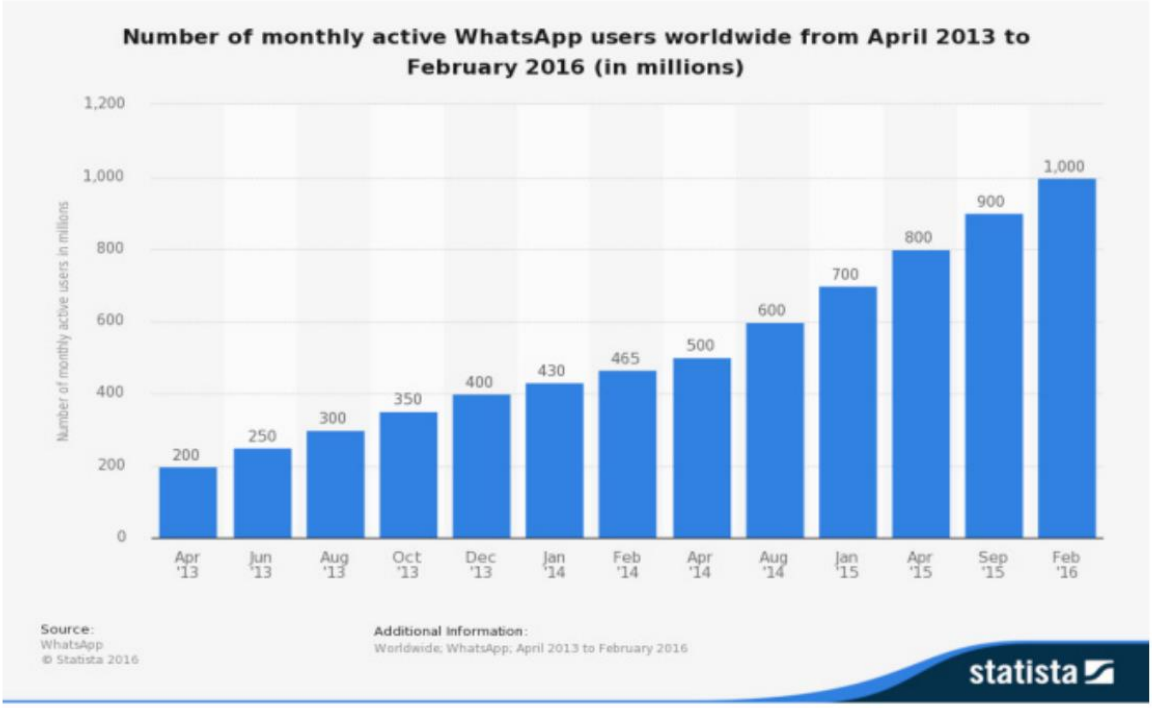
En la figura 1, se evidencia el tiempo en que los usuarios de las aplicaciones móviles pasaron la mayor parte de su tiempo, en redes sociales, mensajes y videos. La mayor penetración para redes sociales es América del Norte con un 60% para la población que tiene por menos una cuenta social. Pero en Estados Unidos el número de penetración llega al 78%, con un promedio general de 109 minutos al día en aplicaciones móviles.

Figura 1. Tiempo promedio gastado al día en aplicaciones móviles



Fuente: Tao, K. (2018) Compilation of Data from International Third-Party Websites and Reports, Including Online Statistics, Market Research, Business Intelligence, and Journal Publications Dated from 2013 to 2017

Figura 2. Número de usuarios activos de WhatsApp hasta febrero de 2016



Fuente: http://cort.as/-Tg0b .

“En un escenario tan tecnológico, el mundo está cambiando de una forma muy dinámica, debido al avance en la tecnología móvil”.[1] Hoy en día es casi imposible no utilizar las llamadas aplicaciones móviles, para comunicarnos con el entorno, esto debido a que muchas personas dependen de estas aplicaciones para sus actividades diarias.

WhatsApp lleva algún tiempo en el mercado, sin embargo, con las actualizaciones constantes en seguridad y servicios, han mejorado mucho su funcionalidad.

Con la evolución del internet, y su velocidad de transmisión ha permitido que los dispositivos móviles tengan un gran avance tecnológico, aumentando la demanda de aplicaciones de mensajería instantánea acortando las distancias entre los usuarios que se encuentren en cualquier parte del mundo. Estos equipos han ido Aumentado su capacidad de almacenamiento, Permitiendo la sincronización y el intercambio de grandes volúmenes de información (audios, videos, documentos) con otros dispositivos, Haciendo de la mensajería instantánea una herramienta que dinamiza las economías de mercado de las regiones, disminuyendo los tiempos de respuesta en las solicitudes de información.

[1] KUMAR, Naveen; SHARMA, Sudhansh. Análisis de encuestas sobre el uso y el impacto de Whatsapp Messenger. *Revista Global del Sistema de Información Empresarial*, 2017, vol. 8, no 3, p. 1

La mensajería instantánea, ha pasado de ser de un servicio exclusivo de computadores a ser una necesidad de comunicación de los usuarios, con el entorno que lo rodea, además de los entornos empresariales que están utilizando estas aplicaciones para mejorar la comunicación con sus empleados y con los entornos empresariales.

Con el aumento en el intercambio de mensajes entre los millones de usuarios, a través de estas aplicaciones han hecho que los datos que viajan por la red sean apetecidos y vulnerables ante los delincuentes informaticos, que siempre están buscando la forma de robar información valiosa. Cuando un dispositivo móvil se pierde o es hurtado, este posee información que puede ser accesada por cualquier individuo, poniendo en peligro la intimidad y confidencialidad de los datos almacenados en estos dispositivos.

Es por estas vulnerabilidades (acceso no autorizado, o pérdida del dispositivo) que se ha pensado en desarrollar un prototipo que haga una encriptación de los mensajes que van a hacer enviados a través de las aplicaciones de mensajería instantánea a otros usuarios, en donde la interacción entre dos usuarios que tengan una clave privada de acceso pueda encriptar y desencriptar el mensaje y poder leerlo.

1. TÍTULO

Diseño de un prototipo de aplicación móvil para cifrado de mensajes sobre plataforma Android.

2. DESCRIPCIÓN DEL PROBLEMA

Con el creciente uso de las aplicaciones de mensajería instantánea, existe la posibilidad de poner en riesgo que se divulgue información confidencial almacenada en los dispositivos móviles. Un delincuente informático puede obtener las contraseñas, información de configuración del sistema operativo, además de archivos personales que se comparten a través de la mensajería. El daño que se presenta por la divulgación de la información robada puede ser mayor que el mismo hurto.

Aunque algunas aplicaciones móviles de mensajería implementan el cifrado extremo a extremo, o punto a punto, almacenando la información cifrada en el dispositivo móvil y no en los servidores de la aplicación, esto que significa, que el cifrado de extremo a extremo evita que terceras personas o la aplicación tengan acceso al texto plano de las llamadas o de los mensajes.

El cifrado de extremo a extremo permite que Solamente el emisor y el receptor puedan leer los mensajes, almacenando las claves de cifrado en el dispositivo de cada usuario, esto además se combinaba con "TextSecure"[2] que emite una clave nueva por cada mensaje enviado, logra que estos mensajes no puedan ser interceptados. Una de las fallas que se han encontrado es que cuando se envía un mensaje y el destinatario no está disponible o no conectado, este mensaje queda almacenado en los servidores de la aplicación, es en este momento que el sistema podría generar una nueva clave para acceder al mensaje. De otra parte, los delincuentes informáticos no siempre atacan la criptografía para romperla, sino que la evitan, para poder tener acceso a los mensajes.

2.1 FORMULACIÓN DEL PROBLEMA

¿Cómo se pueden evitar las fallas en la seguridad en las aplicaciones de mensajería que funcionan sobre la plataforma Android?

[2] Sistema complejo de encriptación, que utiliza una contraseña y que al ser descargada automáticamente almacenaba en una BD y los encriptaba los mensajes de un teléfono

3. OBJETIVOS DEL PROYECTO

3.1 OBJETIVO GENERAL

Desarrollar un prototipo de aplicación móvil que permita cifrar y descifrar texto, y que pueda ser usada en aplicaciones de mensajería instantánea sobre plataforma Android.

3.2 OBJETIVOS ESPECÍFICOS

- Analizar los diferentes algoritmos criptográficos simétricos para seleccionar el más adecuado a su uso en plataformas móviles.

- Desarrollar un componente de software sobre Android que permita aplicar el algoritmo seleccionado.

- Desarrollar los elementos del software que permitan evaluar su rendimiento en el sistema operativo.

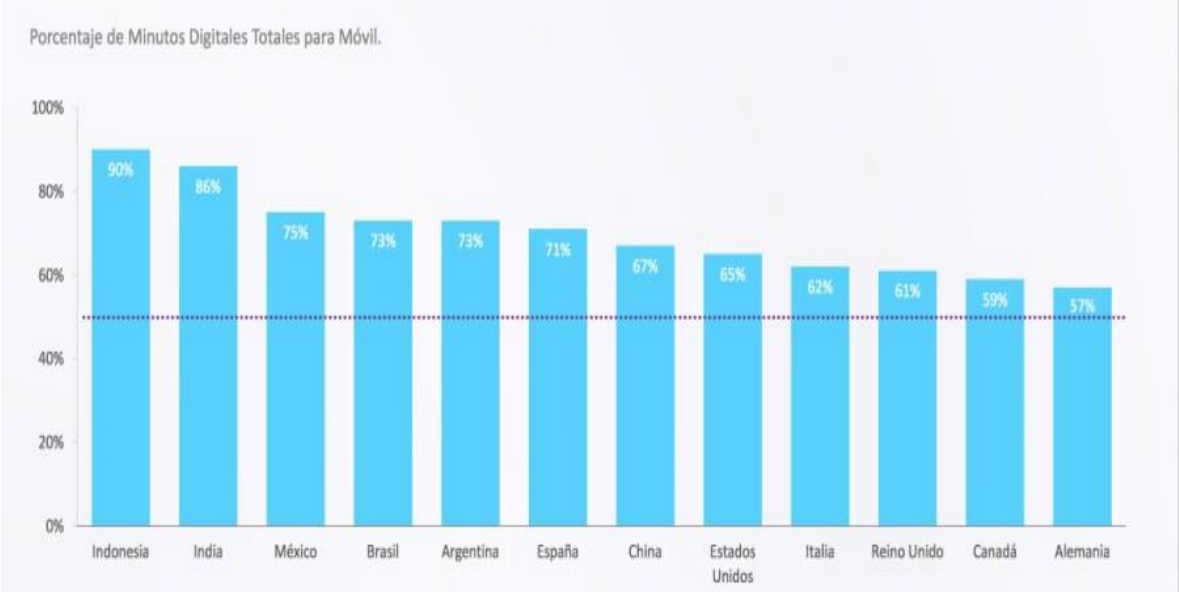
4. JUSTIFICACIÓN

La criptografía en el campo del desarrollo de aplicaciones móviles, es un fragmento importante en el desarrollo de la seguridad, utilizada para el intercambio de información (texto) entre los diferentes usuarios de la plataforma Android, que buscan una mayor confianza y seguridad para la utilización de esta plataforma.

Con el crecimiento en el uso de dispositivos móviles, las aplicaciones de mensajería instantánea se han consolidado como una de las más utilizadas por parte de los usuarios. exponiendo la información almacenada

Desde este punto de vista, el desarrollo de este prototipo busca proporcionar una visión general de las aplicaciones de la criptografía a la protección y seguridad en el intercambio de mensajes de texto.

Figura 3. Porcentaje de consumo de minutos digitales para móviles.



Fuente: https://pickaso.com/2017/estudio-tiempo-uso-dispositivos-moviles

Figura 4. Porcentaje promedio de solo utilización móvil para la audiencia digital

Fuente: https://www.latamclick.com/uso-de-dispositivos-moviles-y-aplicaciones-2017/.

Como se evidencia, en la figura 4 el aumento del uso de tecnologías móviles en los últimos años, ha hecho que se haya creado un mercado para el desarrollo de aplicaciones móviles para diferentes usos y para todas las necesidades. Como es de conocimiento el conjunto de dispositivos móviles que utilizan el sistema operativo Android ha ido creciendo de una forma exponencial, ya que siendo Android de libre acceso permite y facilita la creación de aplicaciones.

La principal motivación para el desarrollo de este prototipo, es que existen pocas, herramientas con cierto grado de seguridad, que permitan encriptar, almacenar y enviar información, como mensajes de texto a través de una aplicación de mensajería instantánea como WhatsApp.

Desde que se creó el sistema operativo Android, se han desarrollado varias modificaciones. Estas modificaciones no muestran información sobre el funcionamiento, sino sobre las escalas de sus API (Interfaz de programación de aplicaciones). Estas escalas se utilizan para tener cierto nivel de conocimiento en la conceptualización, siendo una especie de intermediario entre el usuario y las operaciones que se realizan en el desarrollo de aplicaciones.

A continuación, se evidencian las diferentes API y las versiones del sistema operativo Android.

Tabla 1. Versiones del sistema operativo Android y su API

Fecha de Lanzamiento	Versión Android	API
Septiembre 2008	1.0	1
Febrero 2009	1.1	2
		Cupcake
Abril 2009	1.5	3
		Donut
Septiembre 2009	1.6	4
		Eclair
Noviembre 2009	2.0	5
diciembre 2009	2.0.1	6
enero 2010	2.1	7
		Froyo
Mayo 2010	2.2	
Enero 2011	2.2.1	
Enero 2011	2.2.2	
Noviembre 2011	2.2.3	
		Gingerbread
Diciembre 2010	2.3	9
Febrero 2011	2.3.3	
Abril 2011	2.3.4	
Julio 2011	2.3.5	
Septiembre 2011	2.3.6	
Septiembre 2011	2.3.7	
		HoneyComb
Febrero 2011	3.0	11
mayo 2011	3.1	12
Julio 2011	3.2	13
Septiembre 2011	3.2.1	
		Ice-Cream Sandwich
Octubre 2011	4.0 - 4.0.3	14 y 15
		Jelly Bean
junio 2012	4.1	16
octubre 2012	4.1.1 - 4.1.2	

octubre 2012	4.2	17
Julio de 2013	4.3	18
	KitKat	
octubre de 2013	4.4	19
		Lollipop
noviembre 2014	5.0 - 5.1.1	21-22
		Marshmallow
octubre 2015	6.0 - 6.0.1	23
		Nougat
junio 2016	7.0 - 7.1.2	24-25
		Oreo
agosto 2017	8.0 - 8.1	26-27
		Pie

Fuente: el autor

Algunos problemas presentados en el sistema operativo Android en el desarrollo de sus aplicaciones, están sujetas a la versión del sistema que está instalado. En el presente la gran mayoría de las versiones de este sistema operativo se encuentran trabajando de forma simultánea. El desarrollo de aplicaciones, se puede realizar para muchas de las diferentes versiones de Android, pero no es lo mismo hacer un desarrollo de una aplicación con un API de nivel 22 “que es una versión de Android 5.1 Lollipop que ofrece funciones nuevas a los usuarios y a los desarrolladores de apps” a realizar un desarrollo de una aplicación para un API de nivel 2.

Los dispositivos telefónicos que instalan Android, siguen con la tendencia que los componentes estén integrados en un solo chip. Estas clases de chips se utilizan en sistemas inmersos, que son sistemas desarrollados para realizar una o varias tareas en un sistema en tiempo real. El motivo para que todos los componentes electrónicos se encuentren en un solo chip, hace que se ahorre espacio, haciéndolo muy lucrativo para los dispositivos móviles.

La capacidad que han alcanzado los procesadores instalados en los últimos dispositivos en el mercado de dispositivos móviles, han igualado y en algunos casos superado la capacidad de procesamiento de algunos computadores portátiles.

5. DISEÑO METODOLÒGICO

El prototipo será un producto diseñado, utilizando herramientas de desarrollo para plataformas Android, y mediante la inclusión de librerías que permitan la implementación de manera estándar de algoritmos criptográficos de llave pública y llave privada. Como parte de este proceso se pretende analizar alternativas de algoritmos.

Para la realización de este prototipo se hizo una búsqueda bibliográfica de forma selectiva sobre el tema de investigación. Esta mostró dos niveles de consulta, una desde la seguridad informática y la otra desde el punto de vista particular el criptográfico. Estas búsquedas se realizaron en el buscador de Google con los términos específicos de criptografía y seguridad informática.

La información encontrada fue bastante y variada como se evidencia en la bibliografía de este trabajo, además de gran información en idioma ingles para el desarrollo y orientación del prototipo.

Como primera medida se hizo un comparativo de los algoritmos de encriptación dando como resultado que la mejor opción para el desarrollo del prototipo era la criptografía asimétrica, esto debido que la criptografía simétrica como es el estándar DES (Data Encryption Standard), es un sistema criptográfico que pone como entrada un bloque de 64 Bits del mensaje y le aplica 16 interacciones, sin embargo, utilizado un “método de fuerza bruta”[3], con una prueba de 256 posibles claves se pudo descifrar el algoritmo DES en un tiempo relativamente corto, haciendo inseguro este algoritmo.

Para el desarrollo de este prototipo, se utilizará el “modelo de desarrollo rápido de aplicación (modelo DRA)”[4], teniendo como resultado un prototipo funcional de la aplicación que con posterioridad se irá mejorando en su aplicabilidad. Para el desarrollo de este prototipo su plazo de entrega bastante corto.

En cuanto al software utilizado en la creación de la aplicación, este prototipo se desarrolló en ambiente java, por ser un lenguaje de desarrollo que es independiente de la plataforma en que se esté programando, esto significa que se puede ejecutar el programa desarrollado, en sistemas operativos diferentes (Windows, Linux, Solaris, Android)[5], llamado multiplataforma. Este lenguaje de programación es muy potente y versátil, siendo de código libre y abierto la mayor para trabajar de forma gratuita.

---

[3] Disponible en: https://es.wikipedia.org/wiki/Ataque_de_fuerza_bruta

[4] Anónimo. Desarrollo rápido de aplicaciones (RAD): ¿Qué es y cómo funciona? [En línea]. Disponible en: https://diagramasuml.com/desarrollo-rapido-de-aplicaciones-rad-que-es-y-como-funciona/

[5] Desconocido: https://adictoalcodigo.blogspot.com/2016/07/ventajas-y-desventajas-de-programar-en.html

5.1 TIPO DE INVESTIGACIÓN

El tipo de investigación que se desarrollo fue una investigación de tipo exploratoria.

5.2 METODOLOGÍA DEL DESARROLLO

Son todas las tareas por realizar para llegar al producto final, como son:

- Revisar los algoritmos existentes.
- Análisis del software
- Diseño
- Programación
- Pruebas

6. MARCO REFERENCIAL

6.1 MARCO TEÓRICO

6.2 SEGURIDAD EN ANDROID

Android fue diseñado con seguridad multicapa aprovechando componentes de hardware y software para evitar intromisiones y mantener seguros a los usuarios, esta seguridad proporciona una flexibilidad requerida para una plataforma abierta. Además, no sólo fue desarrollada pensando en la seguridad e integridad del usuario, sino que igualmente también en la seguridad del programador.

A nivel general, Google tiene la capacidad de accesar y eliminar remotamente una aplicación, no sólo de *Android* (*Play Store*), sino también directamente de un dispositivo en dónde este almacenada. La particularidad para la eliminación de una aplicación de forma remota es una intervención de seguridad que Android tiene en donde una aplicación potencialmente peligrosa puede ser eliminada de la circulación activa de forma rápida y segura para prevenir exposiciones a los usuarios. Como dato histórico el control de seguridad fue ejercido por primera vez en junio del año 2010 cuando un investigador de seguridad distribuyó una aplicación de prueba que la podía permitir descargar e instalar otras aplicaciones en el dispositivo.[6]

La aplicación diseñada por este investigador permitía tener control de forma remota sobre el dispositivo en el cual se instalaba. De este modo, el desarrollador de la aplicación tenía acceso a los datos de los celulares de los usuarios que habían descargado dicha aplicación. Se distribuyó como dos aplicaciones distintas, aunque hacían lo mismo.

A nivel del sistema operativo, Android se basa en un modelo de seguridad desarrollado por Linux. Este sistema operativo, tiene como característica principal que cada usuario se identifica con un número, el cual viene con una serie permisos como un grupo.

Cada recurso en Linux tiene asignados tres clases de permisos: como dueño, grupo o público. Los permisos que se pueden dar son: de lectura (read, R), de escritura (write, W) y de ejecución (execute, X).

Cuando se instala una aplicación en Android, se crea un nuevo identificador de usuario o User ID diferente a cualquier otro identificador que se haya creado con anterioridad y la nueva aplicación se ejecutara bajo ese nuevo UID. Todos los datos almacenados por esa aplicación se identifican con esta misma UID, esta es una

[6] Hoog, A. (2011). Android Forensics: Investigation, Analysis and Mobile Security for Google Android.

nueva característica implementada por el sistema operativo Android, debido a que, en Linux, cuando hay múltiples aplicaciones se ejecutan con permisos de usuario. Asimismo, no hay garantía de que este UID sea utilizado por la misma aplicación en otros dispositivos.

Es por ello, que cada aplicación se ejecuta en un proceso diferente y cada ejecución de una máquina virtual separada, las aplicaciones pueden incluir código nativo, que se ejecuta por fuera de la máquina virtual, y se compila para que se ejecute directamente en el procesador del dispositivo Android.

A nivel de aplicaciones las autorizaciones son componentes centrales en la seguridad de Android y no tienen relación con los permisos del núcleo de Linux. Los permisos dan acceso a diferentes servicios como internet, galerías de fotos, componentes de la memoria, acceso a los datos privados del usuario, etc. Cuando un usuario desea descargar una aplicación, este observa en la pantalla del dispositivo los permisos que necesita dicha aplicación, haciendo esto que por parte del usuario conozca el comportamiento y pueda tomar la decisión de que, al instalar la aplicación, esto conlleve a un resultado no favorable.

Con el avance de la información y el crecimiento de las aplicaciones móviles, se ha visto la imperiosa necesidad de proteger los datos personales que se almacenan en los teléfonos inteligentes. Para la protección de esta información la criptografía es el método apropiado para hacerlo. La criptografía se encuentra relacionada con la teoría de la información, seguridad de computadores y ciencias computacionales. (Bibhudendra et al., 2003).

La palabra criptografía proviene del griego Krypto (oculto), y graphos (escribir) traduciendo “escritura oculta”. La criptografía se divide en dos grandes ramas, la criptografia de clave privada o simétrica y la criptografía de llave pública o asimétrica.

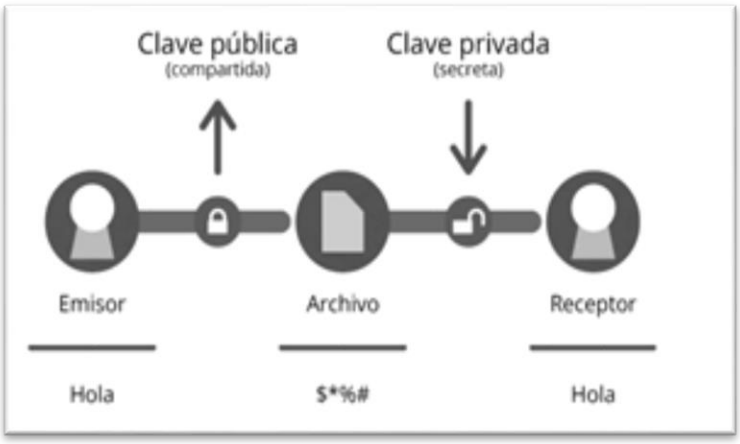
6.2.1 Algoritmos RSA. A mediados de los años setenta dos matemáticos aplicaron algunas fórmulas y conceptos matemáticos para encontrar alguna solución al problema que se estaba presentando con la confidencialidad y autenticidad de la información que se estaba almacenando de forma digital. A este conjunto de técnicas se le llamó algoritmo de llave pública, este es el sistema criptográfico asimétrico más conocido y usado. Con un programa de computación se obtienen un par de números matemáticos relacionados, llamados llaves.

El algoritmo RSA también se puede utilizar para que pueda autenticar un mensaje, cuando se envía un mensaje, el emisor genera un valor hash de dicho mensaje, una vez el receptor recoge el mensaje autenticado, utiliza el mismo algoritmo hash, luego compara el resultado hash obtenido con el valor hash del mensaje enviado. Si ambos coinciden, él sabe que el creador del mensaje tenía conocimiento de la clave secreta del emisor, y que el mensaje no ha sido modificado.

Cuando una persona desea autenticar con un certificado un documento firmado, necesita contar con el archivo y con la llave pública del firmante, esto quiere decir que si un usuario que vaya a autenticar un documento de 10 usuarios deberá poseer los 10 archivos de cada usuario o contar con una base de datos con la información de las 10 llaves públicas de los firmantes. Si llegasen a ser 100 o 1000 posibles firmantes, se crearía un gran problema para la autenticación de estos documentos. Para que se pueda solucionar este inconveniente del manejo de llaves se utiliza el certificado digital, este es un documento que se firma de forma digital y que contiene el nombre del propietario y la llave pública.

Estas dos claves pertenecen a un único usuario que ha enviado el mensaje, una clave es la pública y esta se puede entregar a cualquier usuario, la clave privada es la pública y esta se puede entregar a cualquier usuario, la clave privada debe ser guardada por el propietario de una forma que nadie tenga acceso a ella.

Figura 5. Vista Algoritmo asimétrico



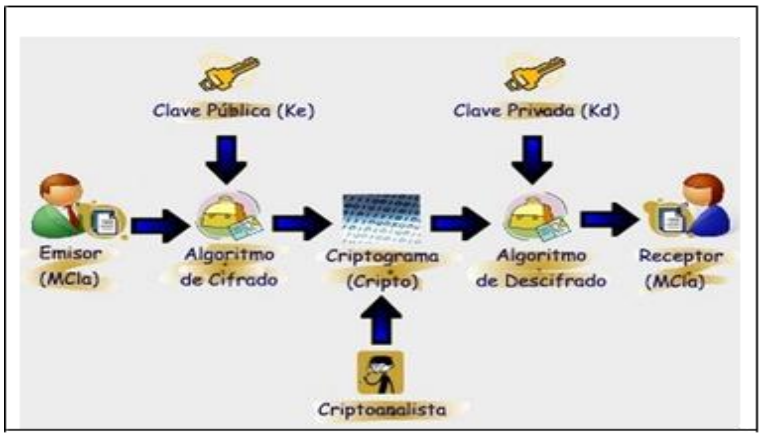
Fuente: http://cort.as/-KGme

El remitente usa la clave pública que envía el destinatario para cifrar el mensaje, una vez cifrado el mensaje, solo la clave privada del destinatario podrá descifrar este mensaje, este es debido a que es el único que la conoce. Por este motivo se logra la confidencialidad en el envío de mensajes, únicamente el destinatario puede descifrar el mensaje.

Este algoritmo se considera seguro, ya que su clave pública se distribuye de una forma gratuita a cualquier individuo que quiera enviar un mensaje, sin embargo, la clave privada jamás se distribuye.

La velocidad con que procesa este algoritmo asimétrico es mucho más compleja que los algoritmos simétricos, puesto que estos requieren mucha más potencia de procesamiento informático, tanto para cifrar como para descifrar

Figura 6. Procedimiento asimétrico



Fuente: http://cort.as/-KGme

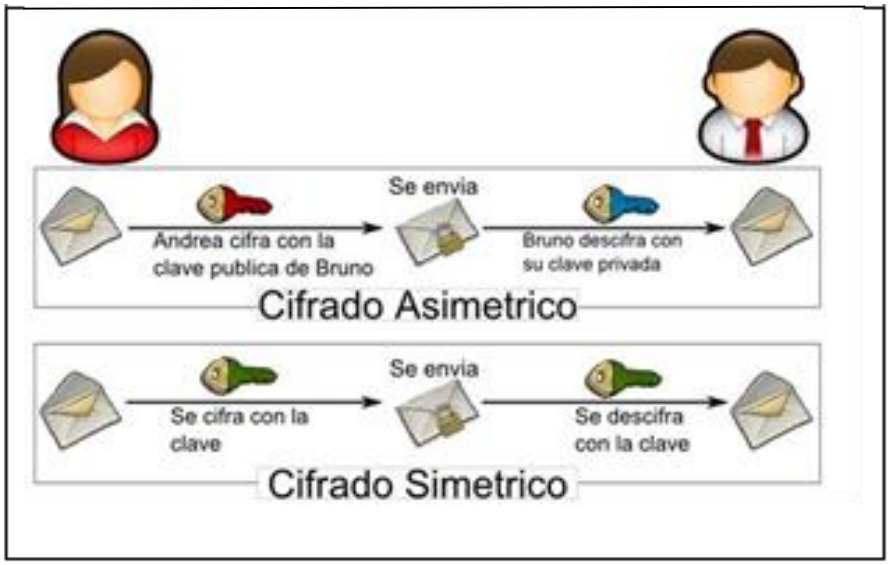
6.2.2 Diferencias entre algoritmos simétricos y algoritmos asimétricos.

Se puede indicar que la criptografía simétrica tiene menos seguridad, esto debido a que, en el momento de transitar por el canal de distribución, la clave crea una gran vulnerabilidad, sin embargo, de todas formas, se puede cifrar y descifrar en un menor tiempo del que se toma la criptografía simétrica. De esta forma se evidencia que la encriptación simétrica mezcla la permutación y la intercalación, y por otra parte, las claves públicas se basan en complejas operaciones matemáticas.

Los algoritmos asimétricos, conocidos como algoritmos de llave pública necesitan mínimo una llave de 3.000 bits, para poder alcanzar un nivel de seguridad similar al que posee el algoritmo simétrico de 128 bits. Estos algoritmos asimétricos son extremadamente lentos, tanto es así que no pueden ser utilizados para encriptar grandes volúmenes de información. Los algoritmos simétricos son aproximadamente 1.000 veces más rápidos que los asimétricos.

Se puede observar en la figura 7; La diferencia entre los algoritmos simétricos y los algoritmos asimétricos, en donde para el cifrado asimétrico se encripta con la clave del emisor y para poder desencriptar, este utiliza la clave enviada por el receptor. Muy distinto a lo que pasa con el cifrado simétrico, en donde se usa la misma clave para su encriptación y su desencriptación.

Figura 7. Diferencias simétricas vs asimétricas



Fuente: http://cort.as/-KGme

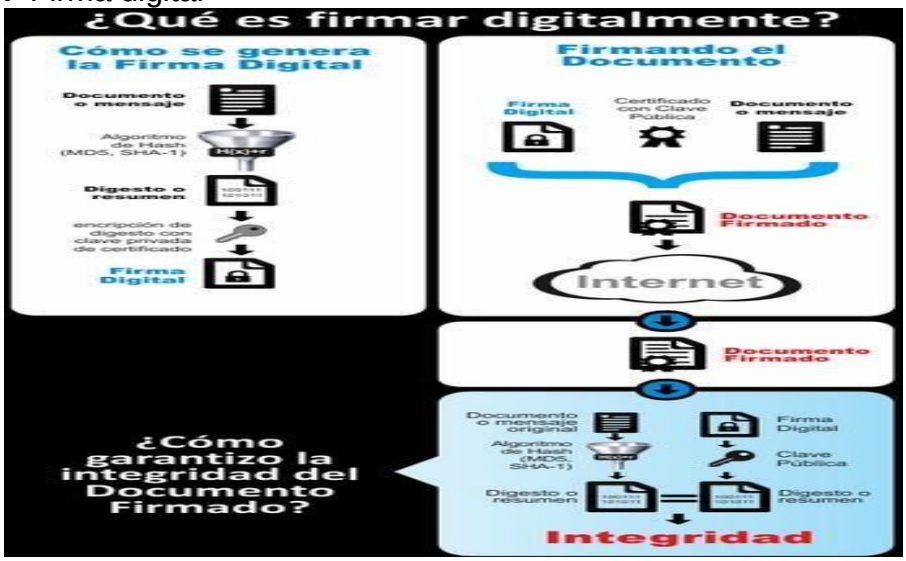
6.2.3 Firma digital. La firma digital es un mecanismo criptográfico avanzado y seguro que está asociado a un mensaje digital que permite cumplir con unos requisitos legales que garantizan la autoría e integridad de los documentos y del mensaje.

Que realiza la firma digital:

- Valida una identidad
- Evita falsificaciones
- Da seguridad a datos confidenciales
- Hace gestiones ante administraciones publicas
- Factura digital.

6.2.3.1 Funcionamiento de la firma digital. Está basado en el uso de dos claves numéricas, una privada y una publica, con una relación entre si matemáticamente que se generan a partir de un certificado digital, que es solicitado por un usuario a una entidad de certificación acreditada. La clave privada debe ser conocida únicamente por su usuario titular y se almacena normalmente en el disco duro o en una tarjeta criptográfica, la clave pública en cambio se distribuye junto con el mensaje firmado. La clave de cifrado solo la tiene una persona, lo que cifra ese usuario podrá ser utilizado como su firma electrónica ya que solo ella puede realizar.

Figura 8. Firma digital



Fuente: http://cort.as/-KGme

Como se observa en la Figura 8. La Firma Digital, se encuentra de una manera resumida y que se realiza primero.

6.2.4 Protocolos de seguridad criptográficos. Este es un protocolo de cifrado abstracto, que realiza unas funciones relacionadas con la seguridad a través de métodos criptográficos.

Por lo general el análisis de los protocolos es difícil, porque las aplicaciones que efectúan dichos protocolos pueden generar problemas adicionales. Por consiguiente, un buen protocolo de seguridad no es suficiente, se debe poseer una buena y robusta implantación de seguridad.

6.2.4.1 Secure socket layer (capa de conexión segura) (SSL). Es una tecnología estándar de seguridad que establece un enlace criptográfico entre un servidor web y un navegador, que proporcionan conexiones seguras por la red en el envío de datos, permitiendo la privacidad e integridad de estos, esto permite la confidencialidad del mensaje.

6.2.4.2 Transport Layer security (seguridad de la capa de transporte) (TLS). Este es un protocolo de seguridad ampliamente adoptado, y diseñado para facilitar la privacidad y la seguridad de los datos en la comunicación en internet. Un uso principal en la utilización de este protocolo es la de cifrar la comunicación entre las aplicaciones web y los servidores (navegadores web). Otra de las aplicaciones que tiene este protocolo es la de cifrar comunicaciones (email, VOIP).

6.2.4.3 Domain Name System Security Extensions (DNSSEC). Es un protocolo de seguridad para moderar el problema de los ataques DNS, protegiendo de estos ataques a través de la firma digital garantizando su autenticidad y de que no han sido manipulados los datos.

Generic Security Services API (servicios de seguridad genéricos API) (GSSAPI). **La característica principal es esta interfase es la de los intercambios** de mensajes opacos a través de los tokens, que generalmente viajan por medio de una red poco segura en donde los mecanismos proporcionan seguridad al mensaje.

6.2.4.4 Hipertext Transfer Protocol (HTTPS). Este protocolo de seguridad es utilizado para el envío de paquetes de datos entre un navegador web y un sitio web. Este protocolo HTTPS se cifra para dar mayor seguridad en la transferencia de datos.

Public key Cryptography standars (Estándares de criptografia de llave publica) (PKCS). **Son un grupo de estándares criptográficos que proporcionan** unos modelos e interfaces de aplicaciones API. Estos estándares o protocolos enfatizan el uso de una llave pública ósea asimétrica.

6.3 ANALIZAR EL DISEÑO DE ALGORITMOS

6.3.1 La criptografía

6.3.1.1 Definición. Es un médico desarrollado para la protección de información y las comunicaciones, mediante el uso de códigos, de tal manera, que solo para quienes está destinada esta información lo podrán leerla y procesarla.

En el área de informática este término se refiere a información segura y técnicas de comunicación a través de conceptos matemáticos y cálculos denominados algoritmos. Estos algoritmos convierten mensajes en formas difíciles de descifrar.

Sistemas de cifrado simétricos. Los sistemas de cifrado simétricos manejan una única clave para el proceso de encriptar y desencriptar los mensajes, Esto quiere decir que las dos partes deben de coincidir en la clave que van a utilizar con anticipación.

Existen dos tipos de operaciones básicas:

6.3.1.2 Cifrado en bloques: los mensajes que se van a cifrar se dividen en bloques con una longitud fija (8,16,32 bytes), después se emplea el algoritmo de cifrado a cada bloque utilizando una clave secreta. Ejemplos: algoritmos DES, AES.

Existen varias formas de operación, esto depende de cómo se mezcla la clave con la información a cifrar, así:

- Modo ECB (Electronic Codebook): El texto se divide en bloques y cada bloque es cifrado en forma independiente utilizando la clave. Tiene la desventaja que puede revelar patrones en los datos.

- Modo CBC (CBC): El texto se divide en bloques y cada bloque es mezclado con la cifra del bloque previo, luego es cifrado utilizando la clave.

- Modos CFB (Cipher FeedBack) y OFB (Output FeedBack): es una modalidad de cifrador en bloques que realimenta el texto cifrado. Para que sea nuevamente cifrado operando el resultado o exclusivo

6.3.1.3 Cifrado de flujo*:* son unos algoritmos de cifrado que pueden hacer un cifrado de forma incremental, haciendo un cifrado de texto bit a bit. Ejemplo: algoritmo criptográfico RC4***.***

Ventajas de los sistemas simétricos:

- Gran agilidad en el cifrado y el descifrado. En tamaño del mensaje no le implica un aumento.
- Su Tecnología es muy difundida y conocida.

Desventajas de los sistemas simétricos:

- La seguridad depende de un secreto compartido entre el emisor y el receptor.
- La administración de las claves no es “escalable".

Un ejemplo claro de criptografía simétrica es el desarrollo de la maquina **Enigma que usaban los nazis para enviar mensajes de sus operaciones en la segunda guerra mundial (**máquina de cifrado mecánico de ejes rotatorios que generaba abecedarios, según la posición de unos rodillos que podrían tener distintas órdenes y posiciones) esta máquina usaba un método simétrico con un algoritmo que dependía de una clave que estaba formada por rodillos que usaba su orden y la posición de cada anillo.

6.3.2 Tipos de algoritmos de cifrados simétricos

6.3.2.1 DES (Data encryption standard*).* El algoritmo DES, es un algoritmo de cifrado en bloque que opera sobre los bloques de texto de 64 bits, devolviendo bloques cifrados también de 64 bits. El algoritmo DES sobre 2 a la 64 posible combinación de bits.

Una de las características del algoritmo DES es que opera con bloques de 64 bits, pero usando tamaños de claves de 56 bits. La clave se almacena usando 64 bits, pero el octavo bit de cada byte (de izquierda a derecha) no se usa.

Esto quiere decir que 8 bits se utilizan como control de paridad (para la verificación de la integridad de la clave). Cada uno de los bits de la clave de paridad (1 de cada 8 bits) se utilizan para controlar uno de los bytes de la clave por paridad impar, es decir que cada uno de los bits de paridad se ajusta para que tenga un número impar de 1, dentro del byte al que pertenece. Por lo tanto, la clave tiene una longitud útil de 56 bits.

El algoritmo se encarga de realizar combinaciones, sustituciones y permutaciones entre el texto a cifrar y la clave, asegurándose al mismo tiempo de que las operaciones puedan realizarse en ambas direcciones (para el cifrado). La combinación entre sustituciones y permutaciones se llama cifrado del producto.
Las partes principales del algoritmo DES son las siguientes:

- Fraccionamiento del texto en bloques de 64 bits
- Permutación inicial de los bloques.
- Partición de los bloques en dos partes: L y R.

Fases de permutación y de sustitución repetidas 16 veces (denominadas rondas) reconexión de las partes izquierda y derecha, seguida de la permutación inicial inversa.

Dado que el algoritmo DES mencionado es público, toda la seguridad se basa en la complejidad de las claves de cifrado.

El algoritmo que a continuación se presenta muestra cómo se debe obtener a partir de una clave de una clave de 64 bits (compuesta por cualquier de los 64 caracteres alfanuméricos), 8 claves diferentes de 48 bits, cada una de ellas utilizadas en el algoritmo DES.

En la primera instancia, se eliminan los bits de paridad de la clave para obtener una clave que posea una longitud de 56 bits.[7]

6.3.2.2 Algoritmo AES (Advanced encryption standard). Es un esquema de cifrado simétrico por bloques desarrollado por dos criptologos de la universidad Leuven en Bélgica. En el año de 1997 fue elegido por el instituto de normas y tecnologías (NIST) como el mejor algoritmo de cifrado. En el año 2003 estados unidos dio el anuncio que este algoritmo era lo suficientemente seguro y que se podría utilizar como un estándar en la protección de la información nacional

Este algoritmo AES hace el cifrado en bloques, con una longitud de bloque variable inicialmente, pero finalmente se define un bloque de 128 bits. Con esto los datos deben ser encriptados en segmentos de 16 bytes (128 bits). Cada uno de estos segmentos se lo puede ver como como un bloque o matriz de 4x 4 bytes. Debido a que es simétrico, la misma clave se utiliza para encriptar y para desencriptar, esta

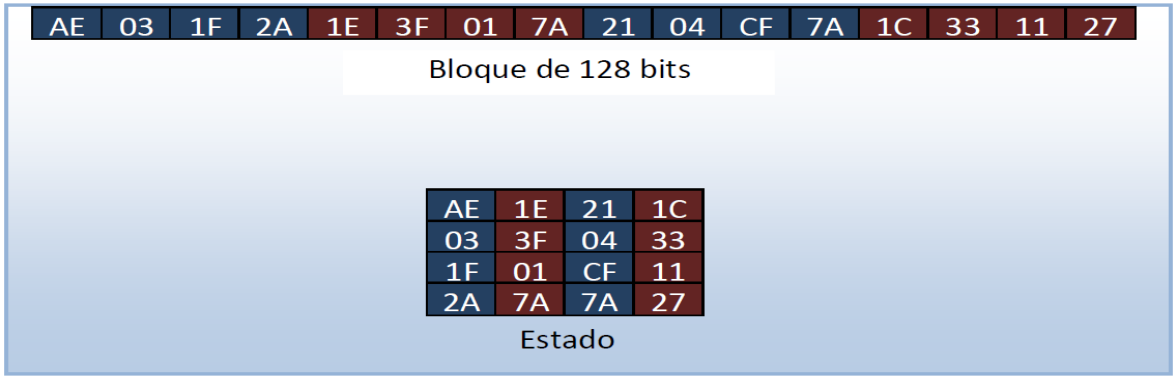
[7] https://es.ccm.net/contents/130-introduccion-al-cifrado-mediante-des

clave puede ser de 128,192, 256 bits según los estándares. Cuando se inicia con una clave de 128 bits, esta genera 10 claves, estas claves que son resultantes además de la clave primaria o inicial se les denomina subclaves.[8]

6.3.2.3 Cifrado AES Operaciones y rondas. Este proceso de cifrado del algoritmo. Este proceso de cifrado del algoritmo reside en aplicar a cada estado un grupo de operaciones las cuales se conocen como rondas que se le aplican a cada estado, en donde el algoritmo realiza once (11) rondas las cuales se clasifican en tres tipos así:

- 1 ronda inicial (que se le aplica a la subclave inicial).
- 9 rondas estándar (se le aplican las 9 subclaves siguientes, una en cada ronda)
- 1 ronda final (se aplica la última subclave).

Figura 9. Aplicación a la subclave inicial



Fuente: http://cort.as/-T8uQ

Las operaciones que realiza el algoritmo dentro de las rondas se reducen a 4 operaciones básicas:

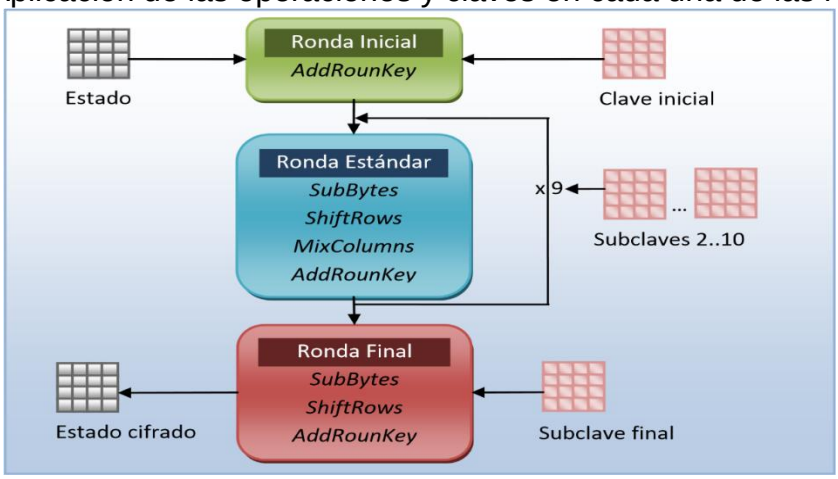
SubBytes.

- ShiftRows.
- MixColumns.
- AddRoundKey.

A continuación, se muestra un diagrama de cómo se aplican las operaciones y claves en cada una de las rondas:

[8] https://revistas.udistrital.edu.co/index.php/Tecnura/article/view/7236/8892

Figura 10. Aplicación de las operaciones y claves en cada una de las rondas



Fuente: http://cort.as/-T8uQ

6.3.2.4 Descifrado AES. El proceso para el descifrado se le aplican las mismas operaciones que se utilizaron para el cifrado, pero se debe hacer de una forma inversa, en donde se utilizan las mismas subclaves que fueron generadas en el orden inverso, además también se utiliza una matriz, que es distinta en la operación, de esta manera se obtendrá la inversa de la transformación lineal aplicada en el proceso para el cifrado.

6.3.2.5 Algoritmo IDEA. Es el mejor y más seguro algoritmo simétrico disponible que existe en la actualidad. Trabaja con unos bloques de 64 bits en su longitud y emplea una clave de 128 bits. Como es el caso de DES, se usa el mismo algoritmo tanto para cifrar como para descifrar.

IDEA es un algoritmo con bastante seguridad, y hasta ahora se ha podido demostrar lo resistente que es a multitud de ataques, entre ellos el criptoanálisis de forma diferencial. Esto es debido a que no presenta claves débiles, y su longitud de clave hace posible en la práctica un ataque por la fuerza bruta. Es considerado por muchos como uno de los cifrados en bloque más seguros que existen.

6.4 LA CRIPTOGRAFÍA ASIMÉTRICA

Este tipo de criptografía está basada en la utilización de dos claves: una clave pública (que se puede divulgar sin ningún inconveniente) y la clave privada (que no se debe revelar en ninguna circunstancia).

Esto puede que parezca a simple vista, que sea un sistema bastante fácil para descifrar, ya que se podría suponer que, si se conoce la clave pública, se podría deducir la clave privada, pero en este tipo de sistemas criptográficos, se utilizan algoritmos demasiado complejos, que generan a partir de la contraseña la clave privada y pública que pueden tener perfectamente un tamaño de 2048 bits.

Ambas pueden ser usadas para encriptar y desencriptar información. Dichas claves están matemáticamente relacionadas entre sí:

- La clave pública está disponible para todos.
- La clave privada es conocida solo por el individuo.

Existen varios algoritmos muy utilizados por ejemplo Diffie-Hellman, RSA, DSA.

Este sistema tiene además dos modos de cifrado:

- Encripción: el mensaje es encriptado usando la clave pública del receptor, el mensaje encriptado es enviado al destinatario, el mensaje recibido se desencripta usando la clave privada del receptor, garantizando así la confidencialidad del mensaje.

- Autenticación: el mensaje es encriptado usando la clave privada del emisor, el mensaje encriptado se envía a uno o más receptores, el mensaje se desencripta usando la clave pública del emisor. Esto garantiza la autenticidad del emisor y la integridad del mensaje.

Este sistema de cifrado tiene las siguientes ventajas:

- No se intercambian claves
- Es una tecnología muy difundida.
- Sus modos cubren los requisitos de seguridad de la información.

Desventajas:

- Requiere potencia de cómputo.
- El tamaño del mensaje cifrado es mayor al del original

6.4.1 Tipos de algoritmos cifrado-asimétricos

6.4.1.1 Algoritmo Diffie-Hellman. Este algoritmo permite pactar una clave secreta para un par de máquinas, esto a través de un canal inseguro y enviando únicamente dos mensajes. La clave secreta resultante no puede ser descubierta por un atacante, aunque se obtengan los dos mensajes enviados a través del protocolo. La principal funcionabilidad de este protocolo es pactar una clave simétrica con la que posteriormente se cifraran las comunicaciones entre dos máquinas.

Actualmente se conoce que es vulnerable a ataques de hombre en el medio (MitM); un atacante podría situarse entre ambas máquinas y acordar una clave simétrica con cada una de las partes, haciéndose pasar por el Host A de cara al Host B y viceversa. Una vez establecidas las dos claves simétricas, el atacante haría de

puente entre los dos hosts, descifrando la comunicación y volviéndola a cifrar para envírasela al otro host.

6.5 MARCO CONCEPTUAL

Algoritmo: es una secuencia de pasos lógicos necesarios para llevar a cabo una tarea específica, como la solución de un problema. Los algoritmos son independientes tanto del lenguaje de programación en que se expresan como de la computadora que la ejecuta. En cada problema el algoritmo se puede expresar en un lenguaje diferente de programación y ejecutarse en una computadora distinta; sin embargo, el algoritmo siempre será el mismo. (Concepto algoritmo, 2019).

Android: es un sistema operativo que está basado en el núcleo Linux. Diseñado principalmente para dispositivos móviles con pantalla táctil, con iconos y una interfaz fácil de manejar. (Ibertronica, 2019).

API: son un conjunto de reglas, códigos y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software. Las API pueden servir para comunicarse con el sistema operativo

Aplicaciones mensajería: La aplicación de mensajería instantánea WhatsApp ha sido una de las aplicaciones que más ha crecido en la utilización a nivel mundial, esto es debido a que los usuarios están creciendo de una forma muy rápida en la utilización de esta mensajería instantánea, tanto en dispositivos móviles, así como en computadores.

Cifrado: es un elemento fundamental de la seguridad de datos y es la forma más simple e importante de impedir la pérdida o robo de la información de un sistema informático con fines mal intencionados. En el mundo de la informática, el cifrado es la conversión de los datos de un formato legible a un formato codificado. (Kaspersky, 2019).

Criptografía: es la ciencia que resguarda documentos y datos que actúa a través del uso de códigos para escribir algo secreto en documentos y datos que se le aplica a la información. (conceptodefinicionde/criptografia, 2019).

FTP: es un protocolo de red, con un conjunto de reglas que establecen como deben comunicarse dos o más entidades para lograr la transformación de la información. Este protocolo esta centrado en la transferencia de archivos a través de una red de tipo TCP/IP que se basa en la arquitectura cliente servidor. (Transmisión Control Protocol). (Pérez Porto, Gardey, 2017).

M2M: (machine to machine o maquina a máquina): es un sistema que permite comunicar una maquina con otra remota en forma de datos para optimizar el

proceso de la función determinada. Todo esto se realiza mediante una serie de elementos que permite gestionar correctamente un envió. La comunicación entre maquinas se realiza mediante el uso de telemática, atraves de diversas redes. (Reporte digital, 2019).

P2P: peer tú peer. Permite la comunicación exclusiva entre dos dispositivos a través de internet con el fin de compartir información. (Internet glosario, 2017)

Prototipo: es un objeto que sirve como referencia para futuros modelos en una misma cadena de producción. Un prototipo es el primer dispositivo que se fabrica y del que se toman ideas más relevantes para la construcción de otros diseños. Por lo general un prototipo no sale a la venta a menos que sea un terminal orientado para que otros desarrolladores de tecnología trabajen con él para insertar nuevas funciones o especificaciones a este para que funcione de una manera más eficiente. (definición,2019).

Root: Hace referencia al proceso que permite obtener privilegios de administrador o súper usuario. Este concepto viene del sistema operativo Linux y se utiliza en Android ya que su base es desarrollado en Linux. Este proceso es necesario cuando se necesita correr ciertas aplicaciones que necesitan privilegios del directorio principal. (Hipertextual, 2015)

TCP: es el protocolo más utilizado en internet. Esta orientado a la conexión, es decir, los datos pueden enviarse de forma bidireccional una vez establecida la conexión. Incluye un sistema automático de comprobación de errores para asegurar que cada paquete es entregado. (speedcheck, 2019).

Telegram: servicio de mensajería por internet, enfocado en la gestión de mensajes de texto y multimedia; inicialmente fue empleado para teléfonos móviles y multiplataforma. (247Tecno,2017)

TextSecure: Sistema complejo de encriptación, que utilizaba una contraseña y que al ser descargada automáticamente almacenaba en una BD y los encriptaba los mensajes de un teléfono. (todotech,2019)

WhatsApp: es una aplicación de mensajería instantánea que se utiliza en teléfonos inteligentes, que puede enviar y recibir mensajes mediante internet, complementando servicios de email, mensajes de forma instantánea, servicio de mensajes cortos o sistemas audios y videos. (Pérez Porto, 2015)

WIFI: Wireless fidelity: es una tecnología de redes inalámbricas que permite la conexión a internet entre diferentes dispositivos, como computadoras, smartphones, mediante el uso de radio frecuencias para la transmisión de la información. (Significados, 2019).

WPAN: Wireless Personal Área Networks- Red inalámbrica de Área Personal: es una red que permite conectar diferentes dispositivos (tantos computadores, puntos de acceso a internet, teléfonos celulares, PDA, dispositivos de audio, impresoras) cercanos a un punto de acceso. Estas redes normalmente son usadas en un rango de pocos metros y de alcance limitado como son los infrarrojos, Bluetooth. (Jiménez, 2019).

6.6 ANTECEDENTES INVESTIGATIVOS

- **Diseño y desarrollo de una aplicación Android para el uso de identidades digitales, autenticación y firmas digitales en sistemas interactivos:** En este proyecto se ha desarrollado una aplicación Android que permite al usuario obtener una identidad, de la cual puede hacer uso, como por ejemplo autenticarse en un sistema. (Blanco delgado, 2014)

- **Desarrollo de una aplicación prototipo para la localización de parqueaderos en la plataforma iOS:** en este trabajo se desarrolla una aplicación para sistemas IOS de Apple, haciendo uso de la metodologia Extreme programming, pasando por cada una de las fases principales: planeación, diseño, codificación y pruebas. (Rojas Cortes, 2013)

- **Diseño e implementación de un software multimedia para el aprendizaje de la criptografia:** en este documento se presenta el desarrollo de una herramienta en donde se muestra las diferentes formas de encriptar mensajes. (Chaves Jiménez, 2008)

- **Rubicon; un nuevo enfoque para la seguridad en las aplicaciones de Smartphone:** en este documento se evalúa mediante el uso de métricas aplicadas en el área de la inteligencia artificial, ampliar el estado del arte en la detección de malware en Smartphone, avanzando en la creación de un entorno más seguro para el uso de este tipo de sistemas. (Borja Sanz, 2012)

- **Análisis y mejora del rendimiento del algoritmo AES para su utilización en teléfonos móviles:** en esta tesis se enmarca en el estudio y desarrollo de algoritmos criptográficos para dispositivos móviles. Se enfoca en la mejora en la implementación del algoritmo AES. (Gálvez Meza, 2014)

- **Descripción poli nominal de los sistemas de cifrado DES y AES:** en este documento se desarrolla la descripción de los sistemas de cifrado de los algoritmos DES Y AES, para distinguir el álgebra conmutativa y la teoría de campos. (García Méndez, 2011)

- **Desarrollo de encriptado AES en FPGA:** en el desarrollo de este trabajo se orienta a la criptografia orientada a la seguridad de las transacciones electrónicas de datos. (Liberatori, 2006)

- **Aplicaciones matriciales a criptografia:** en el desarrollo de este trabajo se evidencia la aplicabilidad que posee la criptografia. (Triana Laverde, 2011)

- **Aplicación Android para la empresa Travelling – service:** este trabajo está orientado al desarrollo de una aplicación para el sector de los viajes, para generar una nueva imagen de marca aprovechando la tecnología. (Gómez Matesanz, 2014)

- **Desarrollo de un chat para dispositivos móviles Android basado en el protocolo de comunicación Bluetooth:** en este trabajo se desarrolla una aplicación que se conecta por bluetooth para la comunicación a través de un chat. (Martínez Coronado, 2012)

- **Diseño y desarrollo de una aplicación móvil para dispositivos Android para un sistema de alerta temprana de los arroyos de la ciudad de Barranquilla, (2014) recuperado:** en este proyecto se presenta la elaboración de una aplicación móvil para el sistema operativo Android utilizando la tecnología del api de Google más, bajo el concepto de patrón de diseño de arquitectura de software MVC (modelo-vista-controlador) y estructurada con la metodologia de programación en cascada. (Raad Licona, 2014)

- **Desarrollo de una aplicación móvil Android para mejorar la integración de los estudiantes de intercambio en la upv mediante uso de herramientas útiles:** en este proyecto se desarrolla una aplicación para la plataforma Android, con lo cual se pretende ayudar a la integración de los estudiantes de intercambio que vendrán a estudiar a la universidad politécnica de valencia. (Perelló soto, 2014)

- **Prototipo de aplicación móvil como herramienta de apoyo para la prevención de riesgos y guía de operación en el acontecimiento de siniestros mediante el uso de realidad aumentada y geoposicionamiento:** con el desarrollo de este trabajo lo que se pretende implementar es una herramienta que permita informar a las personas rápidamente en caso de una emergencia y puedan actuar conforme al plan de prevención. (Roa Montañez, 2015)

- **Desarrollo de una aplicación móvil con cifrado de datos por geo posición:** en este trabajo se desarrolla un contenedor de datos de dispositivos móviles y que además fuesen seguros. (Chicharro Gallego, 2013)

- **Diseño e implementación de un esquema de encriptación y firmas basado en identidad para dispositivos bug:** el tema de esta tesis a desarrollar una alternativa de sistema de seguridad para dispositivos de procesamiento limitado. El sistema utiliza firma de mensajes basada en identidad, pero usando las huellas digitales, tanto para enlazar un usuario especifico a un dispositivo, como para la protección de los datos privados del usuario. (Moreno Vilicich, 2010)

- **Aplicación Android para el control de dispositivos de movilidad usadas en la asociación ASOPLEJICAT:** en este trabajo se aplica en la automatización de una silla de ruedas a través de una aplicación Android para el mejoramiento y autonomía de los usuarios que sufren de impedimentos para llevar una vida normal en condiciones de seguridad y estabilidad. (Rodríguez moya, 2016)

6.7 MARCO LEGAL

Como se está utilizando tecnología para el manejo de información, y esta información podría ser sustraída o robada, se estaría tipificando un delito informático.

Los delitos informáticos son todas las conductas ilícitas susceptibles para sancionar por el derecho penal, cuando se hace un uso indebido de cualquier medio informático.
El delito informático implica actividades criminales, en contra de la confidencialidad, la integridad y la disponibilidad de los sistemas informáticos, redes y datos informáticos, así como el abuso de dichos sistemas, redes y datos.

6.7.1 Legislación global. Regional y local

6.7.1.1 ***Características principales.*** **Son delitos difíciles de demostrar ya que,** en muchos casos, es complicado encontrar las pruebas.

Son actos que se pueden llevar de forma rápida y sencilla. En ocasiones estos delitos pueden cometerse en cuestión de segundos, utilizando solo un equipo informático y sin estar presente físicamente en el lugar de los hechos.

Los delitos informáticos, tienden a proliferar y evolucionar, lo que complica aún más la identificación y persecución de estos.

El uso indebido de los computadores es lo que ha propiciado la necesidad de una regulación por parte del Derecho.

Tabla 2. Los delitos informáticos

DELITO	LEY O NORMA QUE APLICA
Bluesnarfing: es un delito informático, a través del cual alguien se introduce en un teléfono móvil	En España el código penal establece en el artículo 197, penas de uno a cuatro años, para toda

DELITO	LEY O NORMA QUE APLICA
o computador y copia, ve o modifica información. Esta vulnerabilidad aumenta cuando es teléfono está en modo visible y el computador se conecta a redes WIFI libres o abiertas. (delitos informáticos, 2017)	persona que se apodere de datos reservados de carácter personal o familiar de otro, que se hallen registrados en ficheros o soportes informáticos. Las penas serán de dos a cinco años si se difundieran, revelaran o cedieran a terceros los datos. En Colombia la ley 1273 de 2009 determina: art 269A: **acceso abusivo a sistema informático:** el que sin autorización o por fuera de lo acordado, acceda en todo o parte aun sistema informático protegido o no con una medida de seguridad, o se mantenga dentro del mismo en contra de la voluntad de quien tenga legítimo derecho a excluirlo, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 smlv
Obstaculización ilegitima de sistema informático o red de telecomunicaciones: es cuando se impida u obstaculice el funcionamiento el acceso normal a un sistema informático, a los datos informáticos allí contenidos o a una red de telecomunicaciones. Esto se evidencia en ataques DoS "ataques denegación del servicio" (oficina de seguridad del internauta, 2018**)**	**Art 269b:** Incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 smlv, siempre que la conducta no constituya delito sancionado con una pena mayor
Interceptación de datos Informáticos: se considera como la captura de información en movimiento catalogado como un delito de espionaje, como un ataque a la confidencialidad, como accesos a bases de datos y servidores. (ley de protección de datos,2016)	**Art 269 c:** El que sin orden judicial previa intercepte datos informáticos en su origen destino o en el interior del sistema informático, o las emisiones electromagnéticas provenientes de un sistema informático que los transporte incurrirá en pena de prisión de treinta y seis (36) a setenta y dos (72) meses.
Daño informático: el que, sin estar facultado para ello, destruya, dañe, borre, deteriore, altere o suprima datos informáticos o un sistema de tratamiento e información. Este delito puede ser ejecutado por un virus, un gusano, una bomba lógica. (ley de protección de datos,2016)	**Art 269 D:** Incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 smlv
Uso de software malicioso: el que, sin estar facultado para ello, produzca, trafique, adquiera, distribuya, venda, envíe, introduzca o extraiga del territorio nacional software malicioso. Estos softwares son llamados como los keyloggers, es un software que captura las pulsaciones realizadas en un teclado de un equipo para que el delincuente informático pueda conocer las claves de los usuarios de los bancos. (ley de protección de datos,2016)	**Art 269 E:** Incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 smlv
Violación de datos personales: es cuando alguien sin estar facultado para ello, con provecho propio o de un tercero, obtenga, compile, sustraiga ofrezca, venda, intercambie, envíe, compre, intercepte, divulgue, modifique o emplee códigos personales, datos personales contenidos en	**Art 269 F:** Incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 smlv

DELITO	LEY O NORMA QUE APLICA
ficheros, archivos, bases de datos o medios semejantes. Por ejemplo, sustraer información de un computador de una empresa o de cualquier tipo de actividad económica por medio de una usb o cualquier medio de sustracción. (ley de protección de datos,2016)	
Suplantación de sitios web (phishing): cuando se hace la suplantación d un sitio web con una página parecida a la original, busca el posterior robo de información, como son claves de tarjetas de crédito y claves de acceso, estas estafas inician usualmente con un correo electrónico, indicando que la cuenta o usuario podría ser deshabilitada y se deben reingresar datos, re direccionando a través de un enlace falso a una página maliciosa. (ley de protección de datos,2016)	**Art 269 G:** El que con objeto ilícito y sin estar facultado para ello, diseñe, desarrolle, trafique, venda, ejecute, programe o envíe páginas electrónicas, enlaces o ventanas emergentes, incurrirá en penas de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 smlv, siempre que la conducta no constituya delito sancionado con una más grave. En la misma sanción incurrirá el que modifique el sistema de resolución de nombres de dominio, de tal manera que haga entrar al usuario a una IP diferente en la creencia d u accede a su banco o a otro sitio personal o de confianza, siempre que la conducta no constituya delito sancionado con pena más grave.
Hurto por medios informáticos y semejantes: este delito es muy frecuente en los cajeros automáticos, como es el cambio de la tarjeta, la clonación de los datos de la tarjeta por medio de banda magnética en el dispensador del cajero. (ley de protección de datos,2016)	**Art 269 I:** El que, superando medidas de seguridad informáticas, realice conducta señalada en el art 239 manipulando un sistema informático, una red de sistema electrónico, telemático u otro medio semejante, o suplantando a un usuario ante los sistemas de autenticación y d autorización establecidos, incurrirá en penas señaladas en el artículo 240.
Transferencia no consentida de activos: cuando un hacker por medios fraudulento se apodera de información de un usuario y se la vende a terceros para que utilizando esta información compren o paguen de manera virtual con los datos del usuario atacado. (ley de protección de datos,2016)	**Art 269 J:** El que, con ánimo de lucro y valiéndose de alguna manipulación informática o artificio semejante, consiga la transferencia no consentida de cualquier activo en perjuicio de un tercero, siempre que la conducta no constituya delito sancionado con pena más grave, incurrirá en pena de prisión de cuarenta y ocho (48) a ciento veinte (120) meses y en multa de 200 a 1500 salarios mínimos legales mensuales vigentes.

Fuente. El autor

Tabla 3. Legislación internacional de delitos informáticos

PAÍS	NORMATIVIDAD
ALEMANIA	Reforma al código penal (art 148 del 22 de diciembre de 1987) para contemplar los siguientes delitos: Espionaje de datos (202ª) Estafa informática (263ª) Falsificación de datos probatorios (269) Alteración de datos (303ª) es ilícito cancelar, inutilizar o alterar datos e inclusive la tentativa es punible Sabotaje informático (303b)

PAÍS	NORMATIVIDAD
	Destrucción de datos de especial significado por medio de deterioro, eliminación o alteración de un sistema de datos Utilización abusiva de cheques o tarjeta de crédito por lo que se refiere a la estafa informática, el perjuicio patrimonial que consiste en influir en el resultado elaboración de datos por medio de una realización incorrecta del programa, a través de la utilización de datos incorrectos incompletos, mediante la utilización no autorizada de datos
AUSTRIA	Reforma al código penal del 22 diciembre de 1987, se contemplan los siguientes delitos: destrucción de datos (art 126): no solo o datos personales sino también los datos no personales y los programas. Estafa informática (art 148): se sanciona a aquellos que con dolo causen con perjuicio patrimonial aun tercero influyendo en el resultado de una elaboración de datos automática a través de la confección del programa, por la introducción, cancelación, o alteración de datos o por actuar sobre el procesamiento de datos.
CHILE	Chile fue el primer país en Latinoamérica en sancionar una ley en contra de los delitos informáticos. La ley 19223 publicada en el diario oficial el 7 de junio de 1993, señala que la destrucción o inutilización de un sistema de tratamiento de información puede ser castigado con prisión de un año y medio a cinco. Como no estipula la condición para acceder a ese sistema, puede encuadrarse a los autores de virus. Si esa acción afecta datos contenidos en el sistema, la prisión se establecería entre los tres a cinco años El hacking con el ánimo de apoderarse de información, hasta cinco años. Dar a conocer la información almacenada en un sistema puede ser castigado con prisión de hasta tres años.
CHINA	El tribunal supremo chino castigara con la pena de muerte el espionaje desde internet, según anuncio en 2001 Todas las personas implicadas en actividades de espionaje, es decir que" roben, descubran o compren o divulguen secretos del estado" desde la red podrán ser castigados con penas de diez años hasta la muerte. Se consideran actividades ilegales la infiltración de documentos relacionados con el estado, la defensa, las tecnologías de punta o la difusión de virus informático.
ESTADOS UNIDOS	En 1976 dos hechos marcaron un punto de inflexión en el tratamiento policial de los casos: el FBI dictó un curso de entrenamiento para sus agentes acerca de delitos informáticos y el Comité de Asuntos del Gobierno de la Cámara presentó dos informes que dieron lugar a la Ley Federal de Protección de Sistemas de 1985. Esta ley fue la base para que Florida, Michigan, Colorado, Rhode Island y Arizona se constituyeran en los primeros estados con legislación específica, anticipándose un año al dictado de la Computer Fraud y Abuse Act de 1986. Este se refiere en su mayor parte a delitos de abuso o fraude contra casas financieras, registros médicos, computadoras de instituciones financieras o involucradas en delitos interestatales. También especifica penas para el tráfico de claves con intención de cometer fraude y declara ilegal el uso de password ajenas o propias en forma inadecuada. Pero sólo es aplicable en casos en los que se verifiquen daños cuyo valor supere el mínimo de mil dólares. En 1994 se adoptó el Acta Federal de Abuso Computacional (18 U.S.C. Sec 1030), modificando el Acta de 1986. Aquí se contempla la regulación de los virus (computer contaminant) conceptualizándolos, aunque no los limita a los comúnmente llamados virus o gusanos, sino que contempla a otras instrucciones designadas a contaminar otros grupos de programas o bases de datos. Modificar, destruir, copiar, transmitir datos o alterar la operación normal de las computadoras, los sistemas o las redes informáticas es considerado delito. Así, esta

PAÍS	NORMATIVIDAD
	ley es un acercamiento real al problema, alejado de argumentos técnicos para dar cabida a una nueva era de ataques tecnológicos.

Fuente. El autor

En esta tabla se evidencia la legislación internacional sobre los delitos informáticos que aquejan a los diferentes países.

7. PRODUCTO RESULTADO A ENTREGAR

El resultado es un prototipo de aplicación móvil con la característica de cifrado punto a punto, para poder encriptar la conversación que se va a enviar por el chat y que pueda ser desencriptada sin la posibilidad que sea descifrado por el medio de comunicación por el que viaja el mensaje.

El impacto que se espera es que esta aplicación sea de gran utilización por los usuarios de Android para el manejo seguro de sus comunicaciones por redes sociales.

De igual manera se entregarán los manuales de usuario y manual del programador, junto con el código del proyecto.

7.1 LENGUAJE DE PROGRAMACIÓN

En el desarrollo de la aplicación se ira detallando el modo en que se fue desarrollando la aplicación a la cual se le puso como nombre “SMOKE”.

Para el desarrollo del prototipo se utilizará el lenguaje de programación JAVA. Esta es una plataforma informática desarrollada por SUN MICROSYSTEMS, se escogió este lenguaje de programación por ser el más difundido en el medio y al ser muy rápido y seguro.[9] Este lenguaje de programación está orientado a objetos, haciendo que su programación sea más fácil para el pensamiento humano.

7.2 PROGRAMACIÓN

A continuación, se presenta el código fuente del prototipo que se desarrolló para el cifrado de mensajes de texto en plataformas Android:

[9] https://www.java.com/es/security/

Figura 11. Código fuente smoke

```
package org.secure.smoke;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.widget.EditText;

import java.security.Key;
import java.security.KeyFactory;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.SecureRandom;
import java.security.spec.X509EncodedKeySpec;

import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.spec.SecretKeySpec;

public class MainActivity extends AppCompatActivity {
    Key publicKey = null;
    Key privateKey = null;
    static final String TAG = "SmokeApp";
    EditText maintext;
    SecretKeySpec sks = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        maintext = (EditText) this.findViewById(R.id.maintext);
        maintext.setSelectAllOnFocus(true);
    }
    private void generarAES()
    {
        try {
            SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
            sr.setSeed("any data used as random seed".getBytes());
            KeyGenerator kg = KeyGenerator.getInstance("AES");
            kg.init(128, sr);
            byte[] encodedKey = null;
            sks = new SecretKeySpec((kg.generateKey()).getEncoded(), "AES");
            Cipher c = Cipher.getInstance("RSA");
            X509EncodedKeySpec spec = new X509EncodedKeySpec(Base64.decode(maintext.getText().toString(),Base64.DEFAULT));
            KeyFactory kf = KeyFactory.getInstance("RSA");
            Key pubKey = kf.generatePublic(spec);
            c.init(Cipher.ENCRYPT_MODE, pubKey);
            encodedKey = c.doFinal(sks.getEncoded());
            String s=Base64.encodeToString(encodedKey,Base64.DEFAULT);
            maintext.setText(s);

        } catch (Exception e) {
            Log.e(TAG, "AES secret key spec error");
        }

    }
    private void generarRSA() {
        try {
            KeyPairGenerator kpg = KeyPairGenerator.getInstance("RSA");
            kpg.initialize(1024);
            KeyPair kp = kpg.genKeyPair();
            publicKey = kp.getPublic();
            privateKey = kp.getPrivate();
            String s = Base64.encodeToString(publicKey.getEncoded(), Base64.DEFAULT);
            maintext.setText(s);
        } catch (Exception e) {
            Log.e(TAG, "RSA key pair error");
        }
    }
    public void funcionRSA(View v)
    {
        generarRSA();
    }
    public void funcionAES(View v)
    {
        generarAES();
    }
    public void guardarLlave(View v)
    {
        byte[] encodedBytes = null;
        try {
            Cipher c = Cipher.getInstance("RSA");
            c.init(Cipher.DECRYPT_MODE, privateKey);
            encodedBytes = c.doFinal(Base64.decode(maintext.getText().toString(),Base64.DEFAULT));
            sks = new SecretKeySpec(encodedBytes, "AES");

        } catch (Exception e) {
            Log.e(TAG, "RSA encryption error"+e.getMessage());
        }
    }
    public void encriptar(View v)
    {
        byte[] encodedBytes = null;
        try {
            Cipher c = Cipher.getInstance("AES");
            c.init(Cipher.ENCRYPT_MODE, sks);
            encodedBytes = c.doFinal(maintext.getText().toString().getBytes());
            String s=Base64.encodeToString(encodedBytes,Base64.DEFAULT);
            maintext.setText(s);

        } catch (Exception e) {
            Log.e(TAG, "AES encryption error"+e.getMessage());
        }

    }
    public void desencriptar(View v)
    {
        byte[] encodedBytes = null;
        try {
            Cipher c = Cipher.getInstance("AES");
            c.init(Cipher.DECRYPT_MODE, sks);
            encodedBytes = c.doFinal(Base64.decode(maintext.getText().toString(),Base64.DEFAULT));
            String s=new String(encodedBytes);
            maintext.setText(s);
        } catch (Exception e) {
            Log.e(TAG, "AES encryption error"+e.getMessage());
        }

    }
    public void limpiar(View v)
    {
        maintext.setText("");
    }
}
```

Fuente: el autor

7.3 MANUAL DE UTILIZACIÓN DE PROTOTIPO DE APLICACIÓN MÓVIL PARA CIFRADO DE MENSAJES SOBRE PLATAFORMA ANDROID.

Este manual está desarrollado para explicar cómo se debe utilizar el prototipo de aplicación móvil para el cifrado de mensajes:

Para iniciar la descarga de la aplicación, está el enlace disponible:
https://drive.google.com/file/d/1GJiNwoPJpZRfOPpaa44oHrGe-rOVVG/view?usp=sharing

1. Se descarga la aplicación “smoke” en los dos dispositivos que se van a comunicar.

Figura 12. Pantalla dispositivo Android

Fuente: el autor

1. Primero el dispositivo A genera la llave publica y se la comparte al dispositivo B

Figura 13. Menú principal de la aplicación.

Fuente: el autor

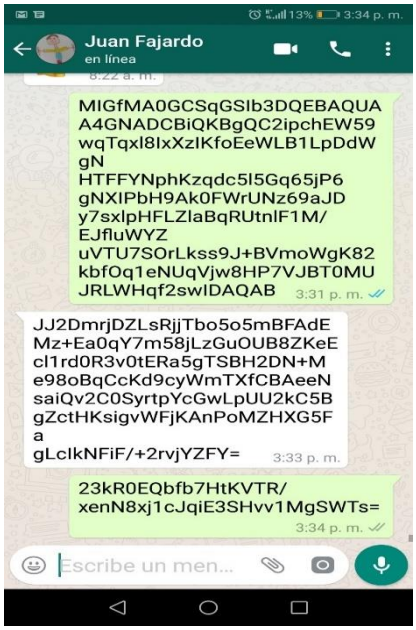
2. Se debe generar la llave publica en alguno de los dispositivos que tenga instalada la aplicación.

Figura 14. Generación de llave publica

Fuente: el autor

2. Se envía la llave publica generada al otro dispositivo

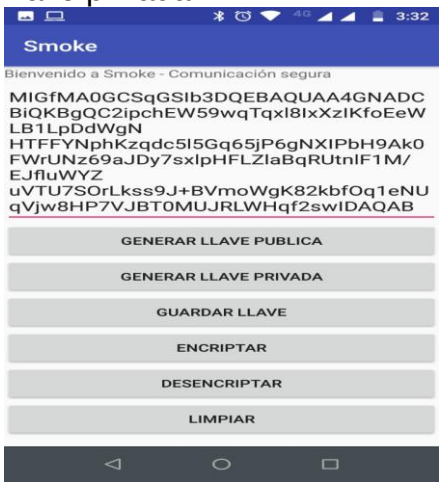
Figura 15. Envió de llave pública



Fuente: el autor

3. Una vez el dispositivo recibe la llave pública la copia y la pega en la aplicación para generar la llave privada, y luego devolverle la llave privada al primer dispositivo.

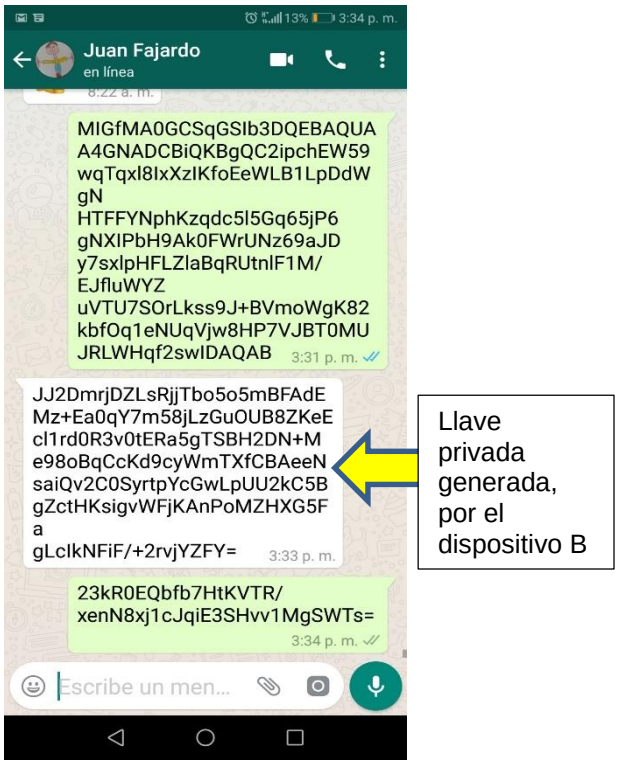
Figura 16. Generación de llave privada



Fuente: el autor

4. Una vez el dispositivo que genero la llave publica recibe la llave privada generada por dispositivo B, la copia, pega y guarda esta llave con la aplicación.

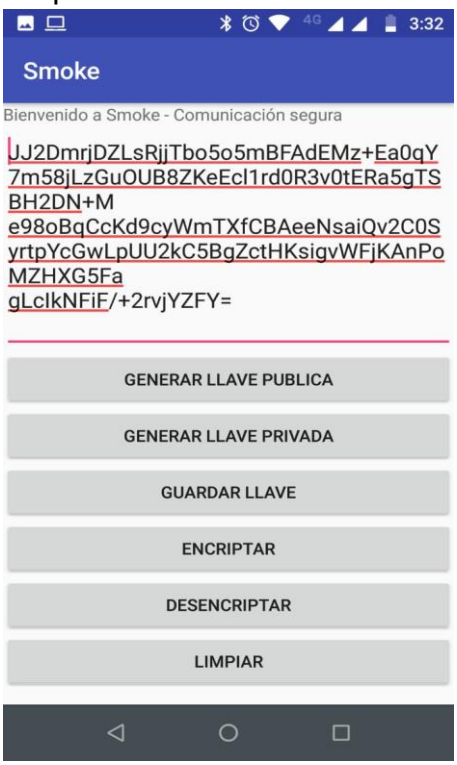
Figura 17. Envío de llave privada



Llave privada generada, por el dispositivo B

Fuente: el autor

Figura 18. Generación llave privada dispositivo A

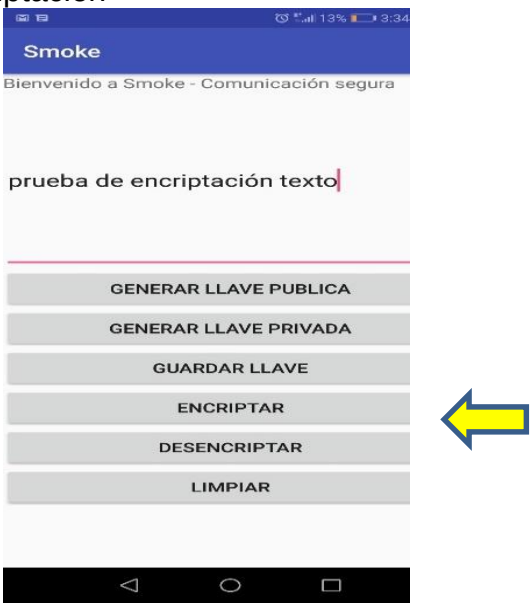


Fuente: el autor

Una vez los dos dispositivos tiene las llaves guardadas, se procede a realizar una prueba de encriptación de un mensaje de texto en la aplicación WhatsApp.

5. Se hace prueba de encriptación de mensaje de texto para ser compartido, una vez se digita el mensaje a encriptar, se da click en encriptar

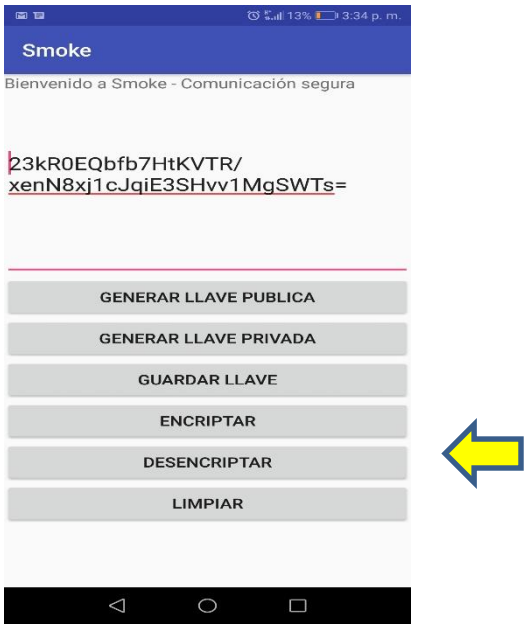
Figura 19. Prueba de encriptación



Fuente: el autor

6. Una vez encriptado el mensaje se copia y se comparte con el dispositivo B

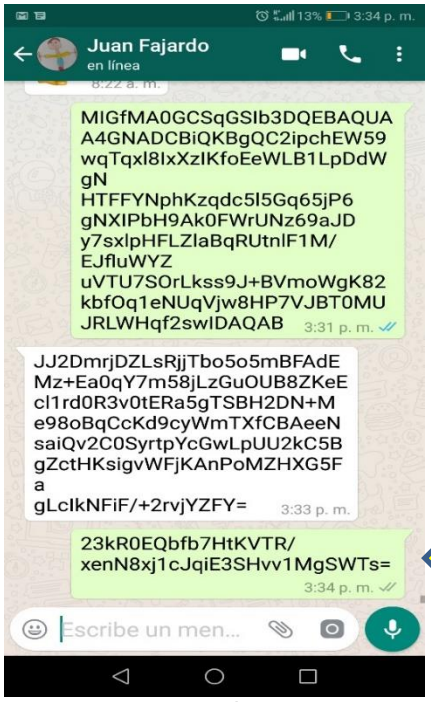
Figura 20. Mensaje encriptado



Fuente: el autor

7. Mensaje enviado y encriptado al dispositivo B

Figura 21. Mensaje enviado

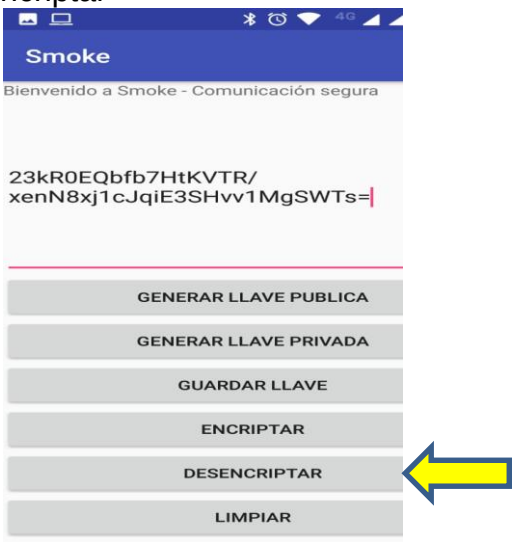


Mensaje encriptado

Fuente: el autor

8. El dispositivo B copia el mensaje encriptado y lo copia en la aplicación smoke y le da la opción desencriptar

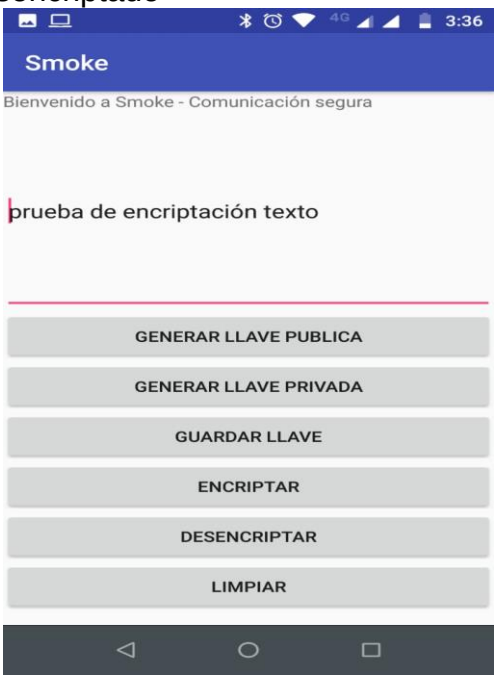
Figura 22. Mensaje a desencriptar



Fuente: el autor

9. Acá se ve en el dispositivo B el mensaje desencriptado

Figura 23. Mensaje desencriptado



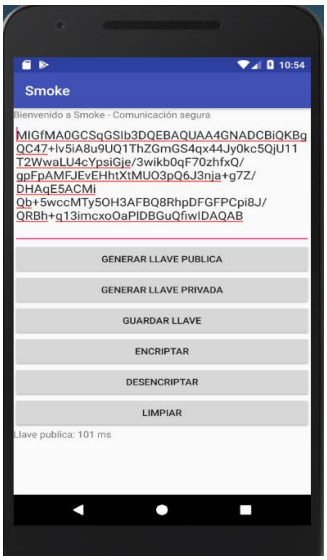
Fuente: el autor

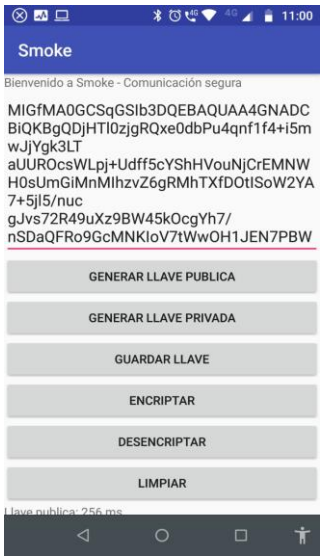
7.4 EVALUACIÓN DE RENDIMIENTO

Se realizó un ajuste en la aplicación para permitir la medición del tiempo de ejecución en milisegundos de las cuatro (4) funciones: generación de llaves, tanto pública como privada, encriptar y desencriptar, determinando lo rápido que se realizan las tareas en el sistema operativo Android en unas condiciones particulares de trabajo o ejecución de la aplicación.

Las pruebas se realizaron en tres dispositivos diferentes como se evidencia a continuación.

Figura 24. Equipos de pruebas





Emulador Nexus 5 (A) Motorola G 5 Plus (B) Huawei P Smart (C)

Características de los equipos de pruebas

Nexus 5:
Es un Smartphone de Google fabricado por LG. Posee una pantalla 1080p de 4.95 pulgadas con protección de Gorilla Glass3, cámara de 8 megapíxeles, procesador quad Core Snapdragon 800 2.26 GHz, 2 GB de RAM, 16 GB de almacenamiento interno y corre la versión Android 4,4 KitKat. (smartGSM, 2019)

Motorola G 5 Plus:
Es un Smartphone fabricado por Motorola. Posee una pantalla Full HD de 5,2 pulgadas, procesador octa-core Snapdragon 625 2 GHz, 4 GB de memoria RAM con 32 GB de almacenamiento interno y corre la versión Android 7.0 Nougat. (smartGSM,2019)

Huawei P Smart:
Es un Smartphone fabricado por Huawei. Posee una pantalla de 5.65 pulgadas a 1080x2160 megapíxeles de resolución, procesador Kirin 659 octa-core 2.3 GHz, con 3 Gb de memoria RAM y 32 Gb de almacenamiento interno y corre la versión de Android 8.0 Oreo. (smartGSM,2019)

De esta forma se realizaron diferentes pruebas de rendimiento con los equipos de usuarios finales, obteniendo los siguientes resultados en la ejecución de tareas, medido en milisegundos.

Dispositivo A:

Tabla 4. Emulador

EMULADOR				
# Ejecución	Generación Llave RSA	Generación Llave AES	Encripción	Desencripción
1	278	11	9	12
2	221	7	9	13
3	178	13	10	8
4	101	9	16	6
5	125	4	15	6
Máximo	278	13	16	13
Promedio	180,6	8,8	11,8	9

Fuente: el autor

Dispositivo B:

Tabla 5. Motorola G5 Plus

MOTOROLA G5Plus				
	Generación Llave RSA	Generación Llave AES	Encripción	Desencripción
1	406	29	47	31
2	381	3	35	21
3	641	5	33	22
4	553	4	45	19
5	555	4	38	18
Máximo	641	29	47	31
Promedio	507,2	9	39,6	22,2

Fuente: el autor

Dispositivo C:

Tabla 6. Huawei P Smart

Huawei P Smart				
	Generación Llave RSA	Generación Llave AES	Encripción	Desencripción
1	85	45	51	31
2	56	7	34	28
3	122	4	44	29
4	147	5	49	26
5	252	4	56	20
Máximo	252	45	56	31
Promedio	132,4	13	46,8	26,8

Fuente: el autor

Tabla 7. Pruebas de rendimiento con diferentes textos

Huawei P Smart					
Prueba	tamaño de archivo(texto)	Caracteres	Palabras	Encripción	Descripción
1	301	2013	346	175	86
2	656	4378	735	151	60
3	4691	30578	5169	1052	116
4	992	6531	1094	216	72
5	1813	11907	2006	293	34
Máximo	4691			1052	116
Promedio	1690,6			377,4	73,6

Fuente: el autor, http://www.contadordecaracteres.info/

En todas las pruebas realizadas se evidencia la respuesta en tiempos óptimos permitiendo comprobar la ejecución dentro de los tiempos esperados

8. RECURSOS NECESARIOS PARA EL DESARROLLO

Tabla 8. Gastos directos:

Ítem	Cantidad	valor
Gastos directos	1 ingeniero de sistemas con conocimiento en desarrollo de aplicaciones y seguridad	$4.000.000
Gastos indirectos	1 computador portátil con 6 GB de memoria RAM con procesador Intel Core i5, sistema operativo w10	$3.000.000
Utilidad o beneficio esperado	Desarrollo del prototipo	$1.000.000

Fuente: el autor

9. CRONOGRAMA DE ACTIVIDADES

Tabla 9. Cronograma de desarrollo de actividades

	JULIO				AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE			
Actividad Semana	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Inicio																				
Definición del tema y formulación de objetivos																				
Análisis																				
Recolección de datos																				
Definir marco metodológico																				
Revisión de información bibliográfica																				
Planificación del proyecto y Recursos																				
Análisis de difentes algoritmos criptográficos																				
Diseño																				
Selección de herramienta utilizar																				
Elaboración del prototipo																				
Diseño de pantalla—s																				
Desarrollo																				
Redacción del trabajo escrito																				
Creación de la interfaz de usuario																				
Configuración de la estructura de la aplicación																				
Creación del manual de usuario																				
Pruebas																				
Desarrollo de pruebas funcionales																				
evaluación de rendimiento de la aplicación																				
Presentación																				

Fuente: el autor

10. CONCLUSIONES

- Debido a las limitaciones técnicas y de procesamiento de los dispositivos móviles el uso de algoritmos simétricos para la transmisión segura de mensajes, resulta ser la mejor alternativa al hacer uso de llaves de tamaño menor comparado contra el uso de algoritmos asimétricos.

- Durante el desarrollo del prototipo se evidenció la versatilidad de AES como herramienta de cifrado y la facilidad que proporciona la plataforma Android para incluir dicho algoritmo en los desarrollos.

- El uso de un algoritmo Asimétrico para el intercambio de la llave privada AES proporcionó un componente fuerte en la seguridad al inicio de la conversación entre los endpoints.

- Si bien el proyecto empleó AES como algoritmo Simétrico y RSA como asimétrico, es posible gracias al uso de diferentes librerías, reemplazarlos por algoritmos alternativos (como 3DES y ECDSA) sin modificar la estructura y flujo del proceso.

- La revisión y exploración de los elementos criptográficos utilizados durante el desarrollo del prototipo proporcionan una visión general de las aplicaciones de la criptografía a la protección del intercambio de mensajes de aplicaciones en diferentes arquitecturas de hardware.

BIBLIOGRAFÍA

ANONIMO, “Criptosistemas de clave pública. El cifrado RSA”, {en línea}. {12 de abril de 2017} disponible en http://www.dma.fi.upm.es/recursos/aplicaciones/matematica_discreta/web/aritmetica_modular/rsa.html

ANONIMO, “International Data Encryption Algorithm (IDEA)”, {en línea}. {12 de abril de 2017} disponible en https://www.ecured.cu/International_Data_Encryption_Algorithm_(IDEA)

ANONIMO, “International Data Encryption Algorithm (IDEA)”, {en línea}. {12 de abril de 2017} disponible en https://www.ecured.cu/International_Data_Encryption_Algorithm_(IDEA)

ANONIMO,” Introducción al cifrado mediante Des”, {en línea}. {12 de abril de 2017} disponible en http://es.ccm.net/contents/130-introduccion-al-cifrado-mediante-des

ANONIMO,” Introducción al cifrado mediante Des”, {en línea}. {12 de abril de 2017} disponible en http://es.ccm.net/contents/130-introduccion-al-cifrado-mediante-des

Aplicación Android para el control de dispositivos de movilidad usadas en la asociación ASOPLEJICAT, {en línea}. {12 de abril de 2017}, disponible en http://dspace.uniandes.edu.ec/bitstream/123456789/4248/1/PIUASIS003-2016.pdf

BEJARANO, pablo g. código enigma, descifrado: el papel de Turing en la segunda guerra mundial. En: eldiario. España. 06, febrero, 2014.

BLANCO DELGADO. Eva. Diseño y desarrollo de una aplicación Android para el uso de identidades digitales, autenticación y firmas digitales en sistemas interactivos, Madrid, 2014, 337 p. trabajo de investigación (Ingeniería de telecomunicación). Universidad autónoma de Madrid. Departamento de ingeniería informática. disponible en http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20140519EvaMilagrosBlancoDelgado.pdf

CAMPOS, Javier “El algoritmo de Diffie-Hellman”, {en línea}. {12 de abril de 2017}, disponible en http://www.javiercampos.es/blog/2011/07/22/el-algoritmo-de-diffie-hellman/

CAMPOS, Javier “El algoritmo de Diffie-Hellman”, {en línea}. {12 de abril de 2017}, disponible en http://www.javiercampos.es/blog/2011/07/22/el-algoritmo-de-diffie-hellman/

Capitulo IV. Legislación en diferentes países sobre delitos informáticos, {en línea}, {15 de mayo de 2017}, disponible en: http://www.poderjudicialmichoacan.gob.mx/tribunalm/biblioteca/almadelia/Cap4.htm

CHAVES JIMÉNEZ, Heidi. Diseño e implementación de un software multimedia para el aprendizaje de la criptografia. Bogotá, 2008, 111p. proyecto de grado. Universidad san buenaventura. Facultad de Ingeniería disponible en http://biblioteca.usbbog.edu.co:8080/Biblioteca/BDigital/43336.pdf

CHICHARRO GALLEGO, Jesus. Desarrollo de una aplicación móvil con cifrado de datos por geo posición, Madrid, 2013, 97p.trabajo aplicado. Departamento de informática. Disponible en http://e-archivo.uc3m.es/handle/10016/18303#preview

Diseño e implementación de un esquema de encriptación y firmas basado en identidad para dispositivos bug, {en línea}. {12 de mayo de 2017}, disponible en http://repositorio.uchile.cl/tesis/uchile/2010/cf-moreno_mv/pdfAmont/cf-moreno_mv.pdf

GÁLVEZ MEZA, Nancy. Análisis y mejora del rendimiento del algoritmo AES para su utilización en teléfonos móviles. México, 2014, 173p. Tesis. Instituto politécnico nacional. disponible en http://148.204.210.201/tesis/1404316762511NPGMTesisAn.pdf

GARCÍA MÉNDEZ, Paulo. Descripción poli nominal de los sistemas de cifrado DES y AES, México, 2011, 106p. Tesis. Universidad autónoma metropolitana. Departamento de matemáticas. disponible en http://mat.izt.uam.mx/mcmai/documentos/tesis/Gen.05-O/Garcia-MPS-Tesis.pdf

GUTIÉRREZ, Pedro “Tipos de criptografia, simétrica, asimétrica e hibrida”, {en línea}. {10 de abril de 2017} disponible en http://www.genbetadev.com/seguridad-informatica/tipos-de-criptografia-simetrica-asimetrica-e-hibrida

KUMAR, Naveen; SHARMA, Sudhansh. Análisis de encuestas sobre el uso y el impacto de Whatsapp Messenger. *Revista Global del Sistema de Información Empresarial*, 2017, vol. 8, no 3, p. 52-57.

Ley 1273 del 5 de enero de 2009, protección de la información y los datos, {en línea}. {12 de mayo de 2017}, disponible en: https://www.mintic.gov.co/portal/604/articles-3705_documento.pdf

LIBERATORI, Mónica. Desarrollo de encriptado AES en FPGA, Argentina, 2006, 172p. Tesis. Universidad nacional de la plata. Facultad de informática. disponible en http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Redes_de_Datos/Tesis/Liberatori.pdf

MARTÍNEZ CORONADO, Nicolás. ROCHA JIMÉNEZ, Omar. Desarrollo de un chat para dispositivos móviles Android basado en el protocolo de comunicación Bluetooth, Bogotá, 2012, 59p. Tesis. Facultad de Ingeniería. disponible en http://repository.ean.edu.co/bitstream/handle/10882/4331/MartinezNicolas2012.pdf ?sequence=1

MENDÍVIL, Ignacio. El ABC de los documentos electrónicos seguros. *Disponibleen: http://www. criptored. upm. es/guiateoria/gt_m163a. htm* 2003.

Paredes, Ángel. Estudio sobre el uso de dispositivos móviles y aplicaciones-tendencias y comportamientos 2017 en: https://www.latamclick.com/uso-de-dispositivos-moviles-y-aplicaciones-2017/.

POUSA, Adrian. Algoritmo de cifrado simétrico AES. Aceleración de tiempo de computo sobre arquitecturas multicore. Trabajo final especialista en redes y seguridad. La plata. Universidad nacional de la plata. Facultad de informática.2011. 41 p.

RAAD LICONA, Anuar. Diseño y desarrollo de una aplicación móvil para dispositivos Android para un sistema de alerta temprana de los arroyos de la ciudad de Barranquilla, Barranquilla, 2014, 86p. Trabajo de grado. Facultad de Ingeniería de sistemas. Disponible en http://repositorio.cuc.edu.co/xmlui/bitstream/handle/11323/238/1140848159-1045692006.pdf?sequence=1

ROA, Fabio. MONTAÑEZ, Miguel. prototipo de aplicación móvil como herramienta de apoyo para la prevención de riesgos y guía de operación en el acontecimiento de siniestros mediante el uso de realidad aumentada y geo posicionamiento, Bogotá, 2015, 131p.Proyecto de grado. Facultad de Ingeniería. Disponible en http://repository.udistrital.edu.co/bitstream/11349/2423/1/RoaGarciaFabioAndres2015.pdf

SANZ URQUIJO, Borja. Rubicon; un nuevo enfoque para la seguridad en las aplicaciones de Smartphone, Bilbao, 2012, 230p. tesis doctoral. Universidad de Deusto. Facultad de Ingeniería. disponible en http://ingenieria.deusto.es/cs/Satellite?blobcol=urldata&blobheader=application%2Fpdf&blobheadername1=Expires&blobheadername2=content-type&blobheadername3=MDT-Type&blobheadername4=Content-Disposition&blobheadervalue1=Thu%2C+10+Dec+2020+16%3A00%3A00+GMT&blobheadervalue2=application%2Fpdf&blobheadervalue3=abinary%3Bcharset%3DUTF-8&blobheadervalue4=inline%3Bfilename%3D%22TESIS+DE+Borja_Sanz.pdf%22&blobkey=id&blobtable=MungoBlobs&blobwhere=1344368491454&ssbinary=true

TRIANA LAVERDE, Juan. Aplicaciones matriciales a criptografia. Bogotá, 2011, 55p. Tesis. Ciencias matemáticas. disponible en http://www.bdigital.unal.edu.co/6255/1/Juangabrieltrianalaverde.2011.pdf .

ENLACE DE VIDEO EXPLICATIVO

- https://drive.google.com/file/d/1YZCOA7LoFA-kC-3synVisu-apbH4MTW5/view?usp=sharing